VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Návrh a konstrukce zařízení využívající technologii RFID pro identifikaci závodníků

Design and Realisation of RFID Devices to Identify Competitors

2019 Kamila Chmelařová

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání bakalářské práce

Student: **Kamila Chmelařová**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2601R013 Telekomunikační technika

Téma: Návrh a konstrukce zařízení využívající technologii RFID pro identifikaci závodníků
Design and Realisation of RFID Devices to Identify Competitors

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je vytvoření přenosného zařízení s využitím platformy Arduino pro čtení a zápis na čipovou kartu a následné přenesení dat do počítače pro potřeby skautské hry.

Vypracovaná práce bude splňovat následující body zadání:
1. Seznámení se s vývojovou deskou Arduino.
2. Seznámení se s problematikou technologie RFID.
3. Výběr vhodných hardwarových prostředků pro vlastní realizaci zařízení.
4. Realizace zařízení na platformě Arduino.

Seznam doporučené odborné literatury:

[1] BALANIS, Constantine. *Antenna Theory : Analysis and design.* 3rd edition. United States : Wiley, 2005. 1117 s. ISBN 0-471-66782-X.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Marek Dvorský, Ph.D.**

Datum zadání: 01.09.2018
Datum odevzdání: 30.04.2019

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky 1D16 OSTRAVA

prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 30. dubna 2019 .....

Abstrakt

Cílem této bakalářské práce je návrh a realizace systému s použitím platformy Arduino. Tento systém zajistí identifikaci účastníků závodu a záznam o jejich průchodu přes kontrolní body na trase. Součástí práce je seznámení se s problematikou RFID (Radio Frequency Identification) po teoretické stránce a s příklady možných použití této technologie. Následuje výběr vhodných hardwarových prostředků (RFID čtečka, modul reálného času) pro konstrukci kontrolních zařízení a jejich následná realizace. V poslední části této práce se zaměřím na návrh a tvorbu software pro přesun dat z čipů do počítače a následné vyhodnocení výsledků závodu.

Klíčová slova: Arduino, RFID technologie, čtečka, čip, modul reálného času

Abstract

The aim of this bachelor thesis is to design and implement a system using the Arduino platform. This system will identify the participants of the race and record their passage through the checkpoints on the route. Part of this thesis is to get acquainted with RFID (Radio Frequency Identification) in theoretical terms and with examples of possible use of this technology. Suitable hardware devices (RFID reader, real time module) are selected for the construction of control devices and their subsequent implementation. In the last section of this work author will focus on the design and creation of software for transferring data from chips to a computer. A method for of evaluation of the results of the race will be presented.

Key Words: Arduino, RFID technology, reader, tag, real time module

Obsah

Seznam použitých zkratek a symbolů

AC	–	Alternating Current
AREF	–	Analog Reference
ASK	–	Amplitude Shifting Key
DC	–	Direct Current
DIP	–	Dual In-Line Package
EAS	–	Electronic Article Surveillance
EEPROM	–	Electrically Erasable Programmable Read-Only Memory
GND	–	Ground
I^2C	–	Inter-Integrated Circuit
ID	–	Identification
IDE	–	Integrated Development Environment
IFF	–	Identify Friend or Foe
LED	–	Light-Emitting Diode
MCU	–	Microcontroller Unit
MISO	–	Master In Slave Out
MOSI	–	Master Out Slave In
NFC	–	Near Field Communication
PCB	–	Printed circuit board
PLX-DAQ	–	Parallax Data Acquisition tool
PWM	–	Pulse Width Modulation
RAM	–	Random-Access-Memory
RFID	–	Radio Frequency Identification
RTC	–	Real Time Clock
SPI	–	Serial Peripheral Interface
SQL	–	Structured Query Language
USB	–	Universal Serial Bus
VBA	–	Visual Basic for Application

Seznam obrázků

Seznam tabulek

ÚVOD

Tato bakalářská práce se zabývá návrhem systému pro vyhodnocení výsledků skautské hry Copak je to za vojáka. Konečným produktem bude několik malých zařízení sloužících jako kontrolní stanice na trase závodu. Tyto přístroje budou rozmístěny po hracím poli, což je v našem případě les. Principem těchto zařízení je, že po přiložení čipu se pomocí technologie RFID (Radio Frequency Identification) na tento čip zapíše aktuální čas a ID stanice. Dále je potřeba platforma pro přenos dat z čipů do počítače o průchodech závodníků jednotlivými stanovišti.

RFID je technologie, se kterou se běžně setkáváme v každodenním životě a stále více se hlásí o slovo. Může nám ulehčit život v mnoha oblastech - v průmyslové, ekonomické, sociální i kulturní sféře. Technologii RFID je možné použít jako náhradu dobře známých čárových kódů, různých štítků označujících výrobky a zboží. S její pomocí lze snadno identifikovat osoby a zvířata, automatizovat výrobní a logistické procesy.

V první kapitole je více rozebrána technologie RFID. Jak a kde tato technologie vznikla, jaké jsou potřeba komponenty pro funkční systém a přehled používaných technologií. V této kapitole jsou dále uvedeny i příklady použití RFID.

Následující kapitola popisuje technologii Arduino. Je zde popsána základní deska Arduino Uno. Dále jsou rozebrané možnosti napájení. Jednak jaké jsou možnosti připojení napájení k Arduinu a také jaké zdroje je možné použít. V této kapitole jsou uvedené i aplikace možné úspory spotřebované energie.

Ve třetí kapitole, která pokrývá praktickou část této práce, je popsána hra Copak je to za vojáka. Nachází se zde pravidla hry a popis stanovišť, která se vyskytují v hracím poli. Tato stanoviště jsou dvojího typu - úkolová a probíhací. Také jsou uvedeny požadavky na systém. Navrhovaný systém se bude skládat ze dvou hlavních částí. Jednak ze zařízení ukrytých v lese, která budou zapisovat data na čipy, a dále ze softwaru, který přečte data z čipů a spočítá výsledky. Je zde uvedena i hardwarová realizace zařízení. Jsou v ní popsány jednotlivé komponenty, které jsou zapotřebí pro realizaci zařízení. Mezi potřebné komponenty patří základní deska s procesorem, RFID čtečka, modul hodin reálného času a další. Dále je popsaná funkčnost zařízení. Je uvedeno, jak funguje Arduino i jednotlivé funkce od počátečního nastavení hodin, přes aktivaci Arduina, až po zapsání dat na čip. Následně je popsán software, který zajistí přesun dat z čipů do počítače a poté pomoci vhodného databázového systému budou vyhodnoceny výsledky. Na závěr této kapitoly je provedena kalkulace nákladů za součástky, které jsou použity na prototyp zařízení. Jsou zde uvedeny i možnosti vylepšení.

1 Technologie RFID

Identifikace je v dnešní době všudypřítomná, a to díky mnoha komplikovaným činnostem, jako je čtení bankovních karet, sledování majetku, výběr mýtného, omezený přístup k citlivým datům atd. Tento druh činností lze realizovat například pomocí otisku prstů, čárových kódů, optického rozpoznávání znaků nebo pomoci čipové karty. Radiofrekvenční identifikace (RFID) je technika k dosažení identifikace objektu pomoci rádiových systémů. Jedná se o bezkontaktní komunikaci, obvykle na krátkou vzdálenost. Tento systém lze úspěšně používat tam, kde je potřeba co nejrychlejší a nejpřesnější zpracování informací a zvýšení efektivnosti různých procesů. Informace je možné ukládat do čipů a následně lze tyto informace číst nebo přepisovat pomoci radiových vln. [1]

1.1 Vznik RFID

Během druhé světové války, nezávisle na sobě, hned v několika zemích probíhal vývoj radiolokačních systémů. Radary sloužily jako varovný systém před blížícími se letouny. Jejich obsluha však nebyla schopna rozeznat, jsou-li na radaru útočící letadla nepřítele nebo vlastní letadla vracející se na základnu. [2]

Němečtí technici zjistili, že pokud piloti při návratu na letiště letouny nakloní, dojde ke změně odraženého signálu. Takto nemotorně letci upozorňovali pozemní personál, že se blíží přátelské letouny, nikoli letadla nepřátel (v podstatě tak vznikl první pasivní RFID systém). Pod vedením Roberta A. Watsona-Watta, otce britského radaru, byl v tajném projektu vyvinut první aktivní systém rádiové identifikace, tzv. IFF (Identify Friend or Foe – rozpoznání přítele/nepřítele). Na každé letadlo britského královského letectva byl instalován vysílač. Když vysílač přijal signál z pozemní radarové stanice, začal vysílat signál zpět, a tím se přihlásil jako přátelský letoun. [2]

V průběhu 50. a 60. let 20. století vývoj pokračoval zejména se zaměřením na využití radiofrekvenční energie k identifikaci objektů na dálku. V 60. letech začaly společnosti využívat k ochraně svého zboží tzv. Electronic Article Surveillance (EAS). Pomocí rádiových vln je možné určit, zda bylo zboží v obchodě zaplaceno či nikoliv. Tyto systémy používají čip o velikosti jednoho bitu, který je připevněn na zboží. Při zaplacení položky obsluha na pokladně čip deaktivuje a zákazník může obchod opustit. Pokud však zákazník nezaplatí a pokusí se odejít z obchodu, čtečka instalovaná u východu zaregistruje čip a spustí alarm. [2]

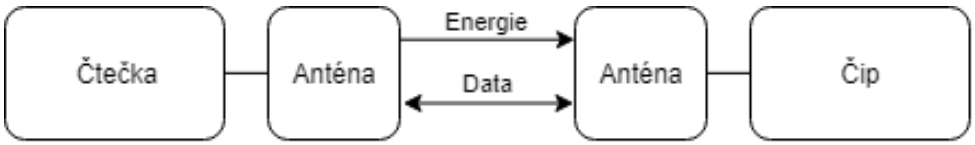
V lednu 1973 si ve Spojených státech Mario W. Cardullo patentoval první aktivní přepisovatelný RFID čip. Ještě ten samý rok si kalifornský podnikatel Charles Walton, původně výzkumník v IBM, patentoval pasivní čip, jenž odemykal dveře bez klíče pouze pomocí ověřování správného identifikačního čísla uloženého na čipu. Jeho nápad odkoupila firma Schlage, která se zámky zabývá a uvedla do praxe doposud využívané přístupové karty. [2]

Následovalo rozšíření technologie a vznik NFC (Near Field Communication). V roce 2004 firmy Nokia, Philips a Sony založily neziskovou organizaci s názvem NFC Forum. NFC technologie byla od počátku vyvíjena jako bezpečná alternativa k technologii RFID, která může být

kvůli svému dosahu často terčem odposlechu. NFC navíc využívá potenciálu mobilních telefonů, které dnes většina lidí nosí a využívá na každém kroku.

1.2 Komponenty systému RFID

Principem systému RFID je uložení dat do paměti radio-frekvenčních čipů a následné opakované čtení nebo zápis dat pomoci čtečky. Čtecí zařízení má podobu pevné brány nebo ručního terminálu. Bezdrátová komunikace probíhá pomoci antény na příslušných radio-frekvenčních vlnách. RFID systém se tedy skládá z čipu a čtečky. Každá tato komponenta má svou anténu. Schéma na obrázku 1.1 je nakresleno v programu draw.io.



Obrázek 1.1: RFID Schéma

1.2.1 Čip

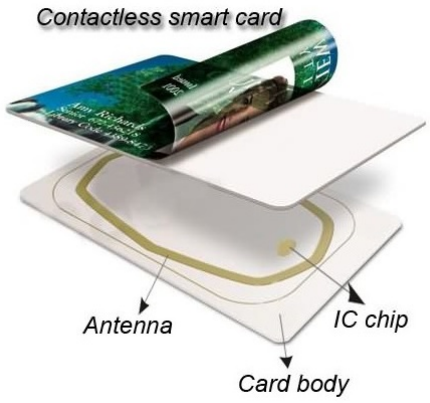
Čipy jsou malá elektronická zařízení, která mohou ukládat data. Označení „transpondér“ vznikl spojením dvou slov, „transmitter“ a „responder“, tedy vysílač a odpovídač. Každý čip má svůj jedinečný kód, pomoci kterého se provádí identifikace. Formy a velikosti čipů se liší podle použití. Čipy se skládají ze dvou nezbytných a dalších volitelných komponentů. Povinné komponenty jsou integrovaný obvod a anténa. Mezi volitelné komponenty patří paměť, senzory sledující teplo, vlhkost apod. Integrovaný obvod se skládá z mikroprocesoru a paměti. Kapacita paměti se liší v různých provedeních.

Anténa v čipu se používá pro komunikaci s čtečkou pro přenos dat a získání potřebné energie pro provoz. Schopnost antény závisí na použité frekvenci, velikosti a poloze antény. Ve většině případů by anténa čipu měla mít všesměrové nebo polokulové pokrytí. Anténa je zatočený drát ve tvaru cívky v obalu čipu. Příklad RFID karty je uveden na obrázku č. 1.2. [3]

1.2.2 Čtečka

Čtečka RFID je zařízení, které využívá rádio-frekvenční vlny k bezdrátovému přenosu dat mezi sebou a čipem. Čtečky mohou být rozděleny do dvou typů: skenerů a pevných čteček. Pevná čtečka je obvykle připevněna na určitém místě. Vzhledem k tomu, že pevné čtečky jsou větší a složitější, mohou být použity pro rychleji se pohybující systémy, jako je například systém klasifikace dopravníků. Na druhé straně je přenosný kapesní snímač.

Mezi hlavní funkce antény také patří dodávání energie pro pasivní RFID čipy. Jakmile se totiž čip vyskytne v oblasti dosahu čtecího zařízení, tedy radio-frekvenčních vln, indukuje se v jeho cívce elektrický proud.



Obrázek 1.2: RFID karta [4]

1.3 Přehled technologií používající standard RFID

1.3.1 Rozdělení podle frekvenčního pásma

Systémy RFID pracují s různými frekvencemi, jejich stanovení vychází z prvotní fáze analýzy řešení. Je třeba určit požadavky na rychlost čtení a zápisu, dosah signálu, prostor pokrytí atd. V tabulce č. 1.1 je uvedeno rozdělení.

Tabulka 1.1: Rozdělení systémů dle frekvenčního pásma [5]

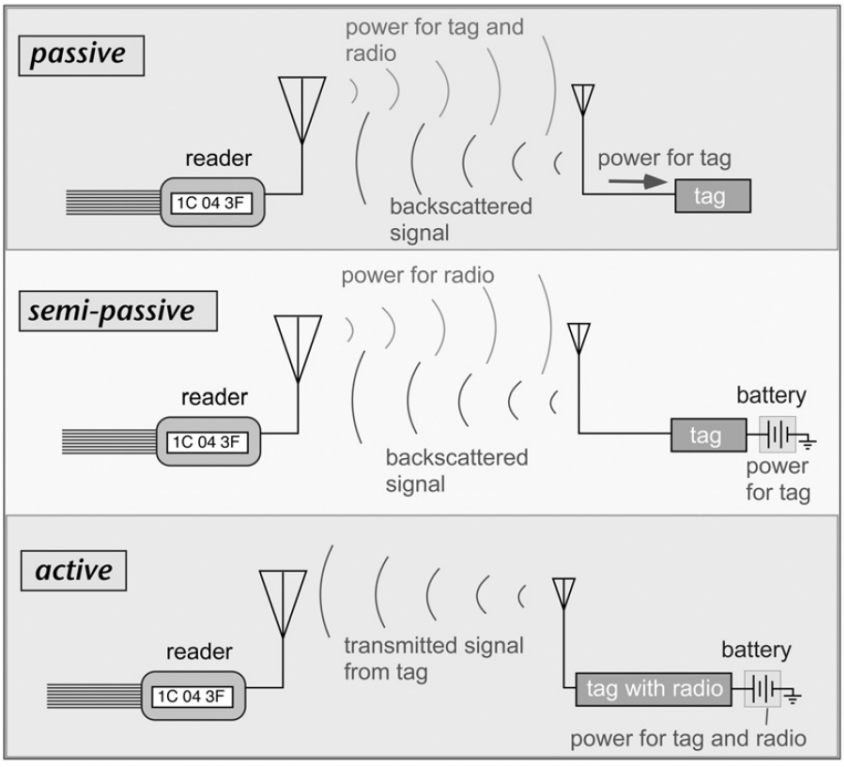
Frekvence Popis	Dosah
125 – 134 kHz (LF – nízká frekvence) platnost celosvětově, v ČR rozšířená frekvence; možnost snímání v blízkosti kovu a přes vodu; nízká rychlost snímání	max. 0,5 m
13,56 MHz (HF – vysoká frekvence) platnost celosvětově, v ČR rozšířená frekvence; obtížné snímání přes vodu; rychlost snímání / zápis cca. 10x rychlejší než LF (20 kB/s)	max. 1 m
865 – 869 MHz (UHF – velmi vysoká frekvence) UHF platné pro Evropu; nelze snímat přes kapalinu a obtížně přes kov	max. 3 m
902 – 928 MHz (UHF) UHF platné pro USA, Kanadu a Mexiko	max. 3 m
950 – 956 MHz (UHF) UHF platné pro Asii a Japonsko	max. 3 m
2,45 GHz ; 5,8 GHz (MW – mikrovlnná frekvence) možnost čtení při velmi vysokých rychlostech; vysoká cena čipů	max. 10 m

1.3.2 Rozdělení podle možnosti zápisu

Další dělení RFID čipů je možné podle toho, zda se jedná o tzv. read-only čipy (bez možnosti zápisu) nebo o tzv. read-write čipy (s možností zápisu). **Read-only** čipy jsou označeny svým jedinečným identifikačním číslem již během výroby nebo během jejich aplikace na zboží. Od toho okamžiku je veškerá komunikace mezi čipem a čtečkou pouze jednosměrná, čtečka nemá možnost na čip uložit další informace, případně změnit informace stávající. Oproti tomu **read-write** čipy obsahují dodatečnou paměť sloužící k uložení informací zaslaných čtecím zařízením. Velikost této paměti může být různá od pár bajtů po několik set kilobajtů. V budoucnu je možné očekávat použití i větších pamětí. Rozhodování o tom, jaký typ čipu použít, záleží na způsobu použití RFID systému. V některých případech je dostačující, aby byl čip nosič informace o jednoznačné identifikaci, jindy je výhodnější mít možnost na čip uložit i další informace, například informace o údržbě výrobku označeného čipem. Je jasné, že čipy bez možnosti dalšího zápisu jsou levnější než čipy s možností zápisu. [6]

1.3.3 Rozdělení podle zdroje napájení

RFID čipy se dělí na aktivní, pasivní a semiaktivní. Hlavním rozdílem je cena, protože v aktivním čipu je zabudovaná i baterie.



Obrázek 1.3: Rozdělení podle zdroje napájení [7]

Pasivní systémy RFID používají čipy bez vnitřního zdroje energie a místo toho jsou poháněny elektromagnetickou energií přenášenou z RFID čtečky. Pasivní čipy se používají pro systémy, jako je kontrola přístupu, časování závodů, smart štítky a další. Aktivní systémy RFID používají čipy napájené bateriemi, které nepřetržitě vysílají svůj vlastní signál. Aktivní značky RFID se běžně používají jako „majáky“, aby přesně sledovaly umístění objektů v reálném čase nebo ve vysokorychlostním prostředí, jako je třeba mýtné. Aktivní značky poskytují mnohem delší čitelný rozsah než pasivní značky, ale jsou také mnohem dražší. Více se zaměříme na pasivní RFID čipy. [8]

Čtečka nejprve vysílá na svém nosném kmitočtu elektromagnetickou vlnu, která je přijata anténou čipu. Indukované napětí vyvolá elektrický proud, který je usměrněn a nabíjí kondenzátor v čipu. Uložená energie je použita pro napájení logických a rádiových obvodů čipu. Když napětí na kondenzátoru dosáhne minimální potřené úrovně, spustí se logický automat či mikroprocesor (tedy řídící obvody uvnitř čipu) a čip začne odesílat odpověď čtečce. Vysílání čipu je realizováno zpravidla pomocí dvoustavové ASK (Amplitude Shifting Key) modulace, která je realizována změnou zakončovací impedance antény čipu. Odrazy, které vznikají změnou impedance antény, jsou detekovány čtečkou a interpretovány jako logické úrovně 1 a 0. Řízení komunikace a jednotlivých stavů komunikačního řetězce je definováno příslušnou ISO normou. [9]

1.4 Využití RFID

Abychom pochopili, jak jsou aplikace RFID rozmanité a jak velký dopad má tato technologie na různé průmyslové odvětví, je zde uvedeno několik příkladů. Záměrně jsou zvoleny zcela odlišné aplikace v různých průmyslových odvětvích, aby byly ukázány téměř neomezené možnosti této technologie.

Nejběžnější použití RFID je v systémech **kontroly vstupu do objektů**. Tento systém je možné využít pro kontrolu přístupu do budov, jako jsou administrativní budovy, průmyslové objekty, školy, bytové nebo rodinné domy atd. Každý pracovník nebo jiná oprávněná osoba má svou identifikační kartu s RFID čipem. Po přiložení čipu ke čtečce dojde ke kontrole, zda daná osoba má povolen přístup do budov nebo místností. Ve firmách je možné tento systém použít i pro kontrolu docházky.

Identifikace hospodářských zvířat je hlavní součástí programů řízení stád. Dříve byla zvířata identifikována ušními štítky, krčními řetězy, mrazícími značkami, tetováním, značkami barvou a mnoha dalšími. RFID čipy mohou být integrovány do ušních štítků nebo umístěny v retikulu, což je sekundární část v žaludku krávy, která zajistí eliminaci rizika ztráty. RFID vede ke zvýšení přesnosti a účinnosti sledování pohybu hospodářských zvířat. Kontrola a sledování nemocí, programy řízení mimořádných událostí, obavy spotřebitelů z hlediska bezpečnosti potravin jsou potřeby průmyslu, které lze lépe zvládnout pomocí značek RFID. Nejdůležitějšími aplikacemi v této oblasti jsou nejen sledování hospodářských zvířat, ale i automatizace krmných stanic a jatek. Další zajímavou aplikací je použití prstenů RFID pro holuby, pro zabránění podvodům. [10]

Klasická metoda **výběrů poplatků za mýtné** spočívala v tom, že každé vozidlo muselo zastavit u mýtné brány a řidič musel zaplatit v hotovosti. Tato metoda vyžaduje výstavbu velkých mýtných bran s potřebou lidské práce a někdy i dlouhé fronty vozidel, což vede k vysokým nákladům. Dnešní vozidla využívající zpoplatněnou silnici jsou vybavena štítkem RFID na vnitřní straně čelního skla. Čtečky na silnicích jsou instalovány buď pod mosty nebo na střešních konstrukcích. Systém dokáže identifikovat vozidla při normálních rychlostech a větších vzdálenostech, která nejsou ovlivněna špatným počasím atd. Tento systém obecně pracuje s aktivními značkami RFID, která jsou stále více nahrazována pasivními značkami. Systém umí i rozpoznat ukradená auta a následně pořídit fotografii.

Vozy vybavené RFID značkami mohou být použity i pro jiné aplikace, příkladem může být parkování. Pomocí RFID čteček instalovanými u vjezdu a výjezdu z parkovacích garáží je možná rychlá a bezhotovostní platba za parkování. [11]

Tuto technologii je možné použít i uvnitř auta. Například společnost SAAB vyrábí klíče s RFID čipem, bez kterého není možné vozidlo nastartovat. Dalším příkladem může být nastavení zpětných zrcátek, polohy volantu nebo sedadla podle toho, kterým klíčem (kterého majitele) se vozidlo startuje. [12]

V **lyžařském průmyslu** také může RFID nabídnout nejrůznější možnosti, hlavně v případě řízení přístupu k lanovce. Bývalá karta s magnetickým proužkem nebo karta vybavená čárovým kódem je stále více nahrazována RFID čipem. Zaplacením zálohy dostane zákazník lyžařský lístek v podobě plastové karty a umístí ji do kapsy. Následně může projít přístupový bod bez nutnosti vytahovat kartu z kapsy.

Ve **zdravotnictví** se mluví o identifikaci pacientů a také o identifikaci zdravotnických předmětů. Zejména ve větších nemocnicích je možné omezit riziko záměny zdravotního zákroku na těle pacienta, případně záměny použití předepsaných léčiv. [12]

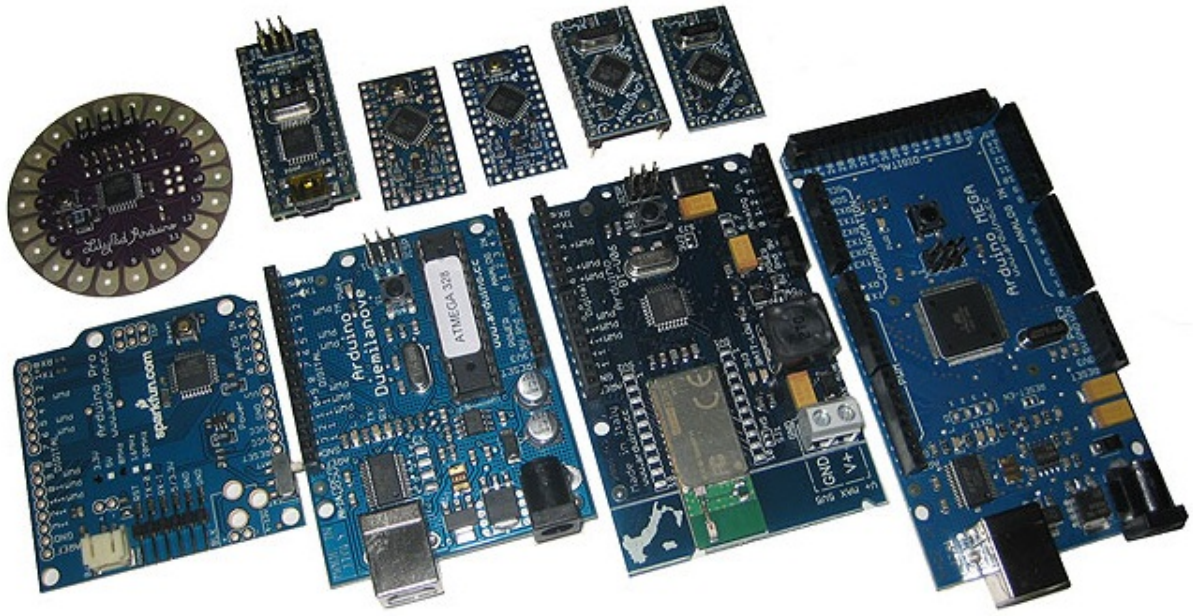
1.5 Výhody RFID technologií

Velmi podobnou technologií, jako RFID, jsou tzv. čárové kódy, které se využívají stále. Přesto má RFID právě proti čárovým kódům několik nezpochybnitelných výhod, které si představíme: [13]

- Bezkontaktní technologie, není potřeba přímá viditelnost,
- odolné RFID čipy (vlhkost, teplota atd.),
- vysoká rychlost čtení, pohybující se většinou v časech pod 100 milisekund,
- aktivní čipy pak přináší nové možnosti interakce do identifikačního procesu,
- v neposlední řadě je to i množství informací, které lze do čipu zapsat, jež se pohybuje kolem 1 MB.

2 Arduino

Termín „Arduino“ zahrnuje jak hardware, tak software. Arduino je prototyp a otevřená platforma založená na snadno použitelném hardwaru a softwaru. Programovatelné desky Arduino zpracovávají vstupy a vytvářejí výstup pomocí programového kódu. K desce Arduino lze připojit několik dalších hardwarových modulů. Open source knihovny jsou dostupné na webu. Díky tomu Arduino může být použito pro širokou škálu projektů od velmi jednoduchých až po velmi složité. Díky tomu je tato platforma velmi flexibilní. Programátoři vytvářejí roboty, dálkově ovládaná vozidla, domácí automatizační zařízení atd. Arduino si díky své jednoduchosti a cenové dostupnosti získalo popularitu. Vzdělávací potenciál lze vidět v několika oblastech, jako je design, elektronika, vestavěné systémy a v neposlední řadě programování. Tyto oblasti se vzájemně doplňují. [14]



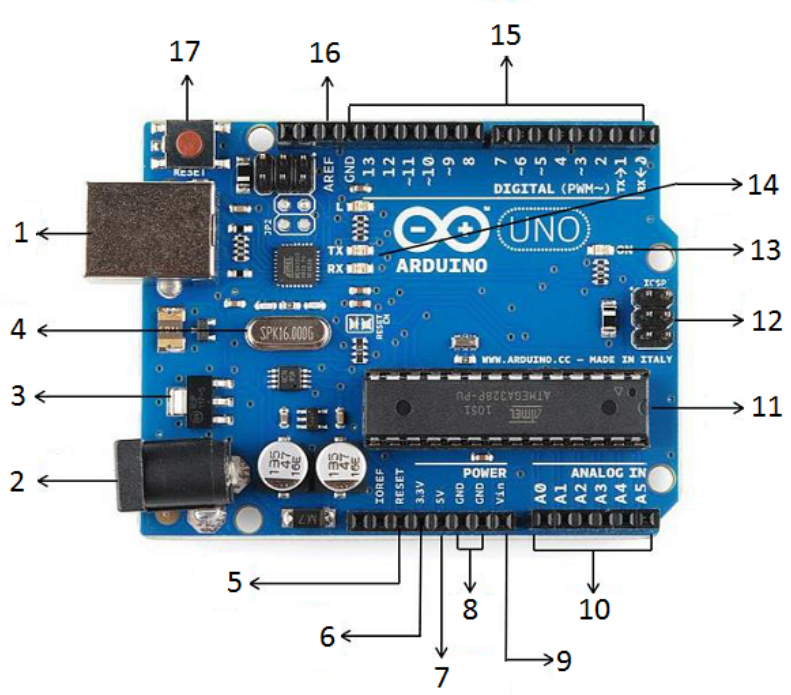
Obrázek 2.4: Základní desky Arduino [15]

Pro programování se používá vývojové prostředí Arduino IDE (Integrated Development Environment). Jazyk, který se používá pro zprovoznění Arduina je C, C++ nebo knihovna Wiring.

Arduino vyrábí od roku 2005 různé typy desek. Při výběru základní desky je důležité si uvědomit cíl a kompatibilnost jednotlivých základních desek. Existují i klony Arduina, jako je ArduPilot, Freeduino, Motoruino, AVRduino, Brasuino atd.

2.1 Popis vývojové desky Arduino

V této kapitole se seznámíme s různými komponenty na desce Arduino. Rozebereme si desku Arduino UNO, protože se jedná o nejčastěji používanou desku. Ostatní desky vypadají sice jinak, ale obsah komponent je podobný. [16]



Obrázek 2.5: Arduino Uno

1. **USB konektor typu B** - Desku Arduino lze napájet pomoci tohoto konektoru přes USB kabel. Tento konektor se používá i pro programování. U nejstaršího modelu najdeme místo USB sériový port. U některých novějších modelů se setkáme s micro USB. Na některých deskách ho vůbec nenajdeme, protože mají buď jiný způsob připojení (Ethernet, BT), nebo pro programování vyžadují připojení externího programátoru.

2. **Napájecí konektor** - Možnost napájení přes souosý konektor ze síťového zdroje nebo z baterie.

3. **Regulátor napětí** - Funkce regulátoru spočívá v řízení napětí na dané desce Arduino a stabilizuje stejnosměrné napětí používané procesorem a dalšími prvky.

4. **Krystalový oscilátor** - Generátor frekvence určující rychlost práce mikroprocesoru. V tomto případě je generovaná frekvence 16 MHz.

5. **a 17. Arduino Reset** - Reset, pokud chceme náš program spustit znovu od začátku. U různých typů Arduina se můžeme setkat i s jiným umístěním tohoto tlačítka. Většinou je ale toto tlačítko popsano nápisem RESET.

6. a 7. Výstupní napájení - Pin „3.3V“ a „5V“ jsou označeny hodnotou pro výstupní napájení. Většina modulů používaných s deskou Arduino potřebuje vstupní napětí 3,3 voltu nebo 5 voltů.

8. GND - (Ground) - Na desce se nachází několik GND pinů, z nichž každý může být použit k uzemnění připojeného modulu.

9. Vin - Tento pin lze také použít k napájení desky Arduino z externího zdroje.

10. Analogové piny - Sem připojíme vodiče, na kterých budeme chtít měřit nějakou analogovou hodnotu. Dají se využít i jako digitální vstupy a výstupy.

11. Hlavní mikroprocesor - Hlavní čip celé desky. V různých podobách a typech ho najdeme na všech deskách.

12. ICSP piny - ICSP hlavice pro externí programování hlavního čipu.

13. Identifikační LED dioda - Svítí, když je Arduino připojeno ke zdroji napájení.

14. L, TX a RX LED diody - Dioda s popisem L je často využívaná. Je totiž připojena k výstupu č. 13. S ní se tedy dá vyzkoušet blikání i bez připojené externí LEDky. Některá Arduina ji však vůbec neobsahují. Diody s popisem Tx a Rx blikají, pokud probíhá komunikace přes sériovou linku.

15. Digitální piny - Deska Arduino UNO má 14 digitálních I / O pinů (z toho 6 PWM (Pulse Width Modulation)) výstup. Tyto piny mohou být nakonfigurovány pro práci jako vstupní digitální piny pro čtení logických hodnot (0, nebo 1) nebo jako digitální výstupní piny pro řízení různých modulů, jako jsou LED diody, relé atd. Piny označené „~“ lze použít pro generování PWM.

16. AREF - (Analog Reference) Někdy se používá pro nastavení referenčního napětí (mezi 0 a 5 V) jako horní mez pro analogové vstupní piny.

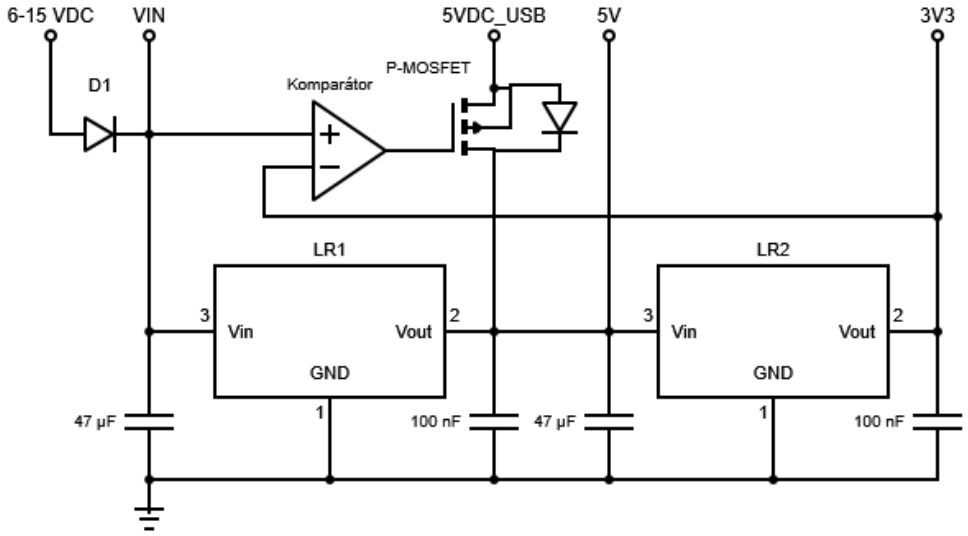
2.2 Napájení vývojové desky Arduino

Arduino je napájeno ze zdroje stejnosměrného napětí, ale je více možností, jak napájení připojit. Nejprve si popíšeme elektrické schéma Arduino, které je na obrázku č. 2.6.

Toto schéma je pro Arduino Uno. Existují i takové vývojové desky, které nedisponují USB portem nebo souosým konektorem. Všechny však mají alespoň pájecí plošky. Mezi jednotlivými typy připojení napájení existují rozdíly, které je důležité zmínit. Obvod Arduina se skládá z těchto prvků:

- Dioda D1, která slouží jako ochrana proti přepólování.
- Lineární regulátor LR1, který generuje na výstupu napětí 5 V DC.

- Lineární regulátor LR2, který generuje na výstupu napětí 3,3 V DC.
- Komparátor a tranzistor typu P-MOSFET, který slouží k automatickému výběru vstupního zdroje napětí („VIN“ nebo „5VDC-USB“). [17]



Obrázek 2.6: Obvod napájení Arduina Uno [17]

Máme tedy tři možnosti, kam je možné napájení na vývojovou desku připojit:

- Souosý konektor,
- dvojice pinů VIN (+) a GND (–),
- USB port.

Napájení přes souosý konektor je ve schématu označeno jako „6–15 VDC“. Na vstupu je dioda D1, která chrání desku proti nechtěnému přepólování. Pravděpodobnost, že dojde k přepólovaní, je malá, protože k souosém konektoru je většinou propojen buď napájecí adaptér, a nebo baterie. [17]

Rozsah vstupního napětí pro připojení přes souosý konektor je 6 až 15 (20 V DC). Ideální je použít napětí v rozsahu 7 – 12 V DC. Maximální proud, který může přes souosý konektor protékat je až 1 000 mA.

Napájení přes pin VIN a GND je velmi podobné jako napájení přes souosý konektor. Na schématu (obrázek č. 2.5) je svorka označena jako VIN. Oproti souosému konektoru není na vstupu žádná ochranná dioda (D1). V případě, když chceme Arduino napájet z baterie, je nejvhodnější použít tento typ napájení pro co nejmenší spotřebu. Na každé diodě je určitý úbytek napětí. V případě diody D1 nastává pokles přibližně o 0,8 V. Tento úbytek má vliv na výdrž baterie. Pro zajímavost, kdybychom z baterie odebírali konstantní proud 0,2 A, tak na

diodě D1 bychom měli ztrátový výkon:

$$Pz = UI = 0,8 * 0,2 = 0,16\ W$$

Dioda D1 (její úbytek 0,8 V) má vliv i na rozsah vstupního napětí. Ten je při napájení přes VIN a GND 5,2 – 14,2 V. Doporučený rozsah je 6,2 – 11,2 V. Proudové omezení je stejné jako u souosého konektoru. [17]

Napájení přes USB port využíváme hlavně při programování Arduina. Napájení přes USB je možné využít i ve finální verzi, kdy budeme chtít zařízení napájet z nabíječky na telefon nebo z powerbanky. Velká výhoda tohoto napájení je, že může být kromě USB zároveň připojen i další napájecí zdroj (souosý konektor nebo VIN+GND). Často nastává situace, kdy je potřeba nahrát novou verzi kódu do Arduina, které už je instalováno ve „finálním produktu". Tento produkt už je nejspíš napájen např. přes souosý konektor, který není nutné odpojovat (resp. vypínat hlavní zdroj). Na závěr uvádím shrnující tabulku. [17]

Tabulka 2.2: Souhrn základních informací o napájení [17]

Typ připojení	**Rozsah napájecího napětí (V)**	**Max. proud (mA)**
Souosý konektor	6 až 15 (20)	1000
VIN+GND	5,8 až 14,8 (19,8)	1000
USB port	4,75 až 5,25	500

2.2.1 Napájecí zdroje

Následuje rozbor možných zdrojů, které se u Arduina používají:

- Síťové zdroje,
- spínané DC-DC měniče (step-up a step-down),
- akumulátory a baterie.

Základním typem zdroje je **síťový zdroj**. Na jeho vstupu jde ze síťové zásuvky napětí 230 V AC, které zdroj převede na výstupu na stejnosměrné napětí menší velikosti. Tyto zdroje můžeme dále rozdělit podle různých kritérií, např. podle výkonu, podle výstupního napětí, stabilizované a nestabilizované, lineární a spínané, dle typu montáže. Při výběru je potřeba se zaměřit na velikosti výstupního napětí (mějme na paměti povolené rozsahy pro Arduino), je vhodné použít např. 9 nebo 12 V DC. [17]

Spínané DC-DC měniče využijeme v situaci, kdy bude potřeba Arduino přidat k již existujícímu zařízení, které má obvykle vyšší napájecí napětí, než je povolený rozsah. Nebo naopak bude potřeba jako zdroj energie pro Arduino použít tužkovou baterii, která má jmenovité napětí 1,5 V DC, což je také mimo povolený rozsah. V takových situacích je možné použít DC-DC

měniče, které umějí převést vstupní napětí na jiné napětí, které už bude v povoleném rozsahu. Existují dva základní typy:

- Step-down měniče,
- step-up měniče.

Step-down (buck) měniče snižují vstupní napětí. Tedy např. z 24 V DC na 12 V DC. Oproti tomu step-up (boost) měniče umějí pravý opak – převádějí vstupní napětí na vyšší. Tedy např. z 1,5 V DC na 9 V DC.

Při sestavování obvodu s měničem napětí je důležité si uvědomit, že zdroj, ze kterého měnič napájíme, musí mít větší výkon, než je příkon zátěže (Arduina), a to ideálně o více než 25 %. Pokud tato podmínka nebude dodržena, zařízení nebude správně fungovat. [17]

Akumulátory a baterie jsou zařízení, které slouží k uchování elektrické energie. Primární článek (baterie) dodává svou energii hned a již jej není možné znovu nabít. Naopak sekundární článek (akumulátor) je třeba nejdříve před použitím nabít a následně je možné opakovat nabíjení. Nejčastěji se u článků uvádí napětí a kapacita. Klasická AA baterie má nominální napětí 1,5 V a kapacitu 3 000 mAh. Z toho lze odvodit, že baterie dokáže napájet zařízení s napětím 1,5 V a proudem 30 mA po dobu 100 hodin. Při vyšším odběru se tato doba zkracuje.

Z uvedeného plyne, že snížením náročnosti na velikost dodávaného proudu se snižuje spotřeba, respektive energetická náročnost zařízení. To se projeví delší životností baterie (samovybíjení se nezohledňuje). [18]

Baterie se vyznačují následujícími charakteristikami:

- Mají omezenou životnost a kapacitu,
- samovolně se vybíjejí,
- potřebují výměnu / dobíjet,
- dodávají stejnosměrné napětí a proud.

Stejně jako u ostatních zdrojů energie i zde je nutné se zaměřit na to, aby jmenovité napětí zdroje bylo v povoleném rozsahu Arduina. Musíme tedy vědět, jaké je minimální a maximální provozní napětí, které může článek mít. Např. u 9V baterie je maximální provozní napětí přibližně 9,6 V DC a minimální cca 6 V DC. Vzhledem k tomu, že tato napětí jsou v povoleném rozsahu pro Arduino, můžeme baterii rovnou připojit ke svorkám VIN+GND. [17]

Mezi kritéria pro výběr vhodného článku patří jmenovité napětí, kapacita, typ (dobíjecí/nedobíjecí), použitý materiál, dostupnost.

2.3 Energetická efektivita

Jedna z velmi požadovaných vlastností dnešních elektronických zařízení je energetická efektivita. Nejvíce je to potřeba u přenosných zařízeních, která jsou napájena z baterií. Baterie má velice omezené možnosti z pohledu kapacity a velikosti dodávané energie.

Elektronické obvody v zařízení se dají nahradit modelem rezistoru, pomocí kterého je možné zjednodušit výpočty. Tímto modelem je možné simulovat měření proudu a napětí. K výpočtu spotřeby energie se používají tyto vzorce:

$$P = I^2 \cdot R \qquad [\text{W; A, } \Omega]$$
$$P = V \cdot I \qquad [\text{W; U, I}]$$
$$V = I \cdot R \qquad [\text{U; A, } \Omega]$$
$$W = P \cdot t \qquad [\text{J; W, s}]$$

kde P je výkon, I je proud, R je odpor, V je napětí, W je práce, t je čas [18]

Důležitý je poslední vzorec, protože ve výpočtu zohledňuje i čas. Při výběru baterie se bere v úvahu vždy výkon za čas.

Princip šetření energie se dá aplikovat i v domácnosti. Při odchodu z domu vypínáme všechna zařízení, jako třeba televizor nebo světlo. Jelikož se nepoužívají, zbytečně by spotřebovávala energii. Stejné je to i v případě elektronických zařízení. Existuje několik možností, kterými je možné snížit spotřebu energie:

- Vypnutí externích periférií,
- úprava frekvence procesoru,
- řetězení mikroprocesorů,
- vypnutí brown-out detekce,
- vypnutí A/D převodníku,
- vypnutí watchdog časovače,
- spící režim procesoru.

Externí periferie, které delší dobu nevyužíváme, je vhodné vypnout. Zapínaní a vypínaní v krátkých časových intervalech by mohlo mít opačný efekt - zvýšení spotřebované energie. Rozhodnout, jaká doba je vhodná pro vypnutí modulu, je na tvůrci zařízení. Mezi externí periferie patří třeba energeticky náročné displeje, externí paměť či I/O piny trvale nastavené na úroveň HIGH. [18]

Snížení frekvence hodinového signálu také způsobí menší spotřebu energie. Nevýhodou je však zpomalení procesoru. Sice procesor spotřebuje méně energie, ale pro výpočty bude potřebovat delší čas. Ve výsledku tedy není jisté, jestli k úspoře dojde. Závisí to na konkrétních hodnotách. Možnou alternativou je optimalizovat kód tak, aby ho procesor zpracoval rychleji. Cena však představuje snížení jeho čitelnosti a snadnosti ladění. [18]

Pokročilejším řešením je vytvoření **režimů procesoru**. Režimy je možné přepínat podle potřeby. V případě, kdy je zapotřebí rychle zareagovat nebo poskytnout výpočty, je procesor v „rychlém režimu“, při čekání na vstup se procesor může přepnout do „pomalého režimu“. Existuje několik režimů spánku s rozdílnou hloubkou spánku, což se odráží i na úspornosti použitého režimu. Jejich základní rozdělení a popis jsou uvedeny v tabulce 2.3. Charakteristiky a principy využití jednotlivých režimů se v závislosti na výrobci liší. Konkrétní řešení je potřeba ověřit dokumentací k použitému mikroprocesoru. Ve všeobecnosti je ale princip režimů vždy stejný. [18]

Tabulka 2.3: Režimy spánku u Arduina [18] upraveno

Režim		Popis
Běžící (Run)	EM0	Standardní režim práce, kdy je procesor během celé doby dostupný a zpracovává přidělené operace
Pohotovostní (Stand-by)	EM1	Režim, který umožní rychlý návrat do aktivního módu díky oscilátoru, který zůstává zapnutý. Probuzení se realizuje na základě přerušení. Návrat do aktivního režimu způsobí krátké zvýšení spotřeby. Průměrná spotřeba je cca 200 μA. Čas potřebný k návratu do aktivního stavu je jeden hodinový cyklus.
Hluboký spánek (Deep sleep)	EM2	V tomto režimu zůstávají aktivní jen kritické části - LCD ovladač, sběrnice či watchdog časovač. Obsah RAM a registrů CPU je uložen. Průměrná spotřeba je mezi 10 až 50 μA. Čas potřebný k návratu do aktivního stavu je 5 až 8 μs.
Zastavený (Stop)	EM3	Hlubší režim spánku. Ukazatele a konfigurační registr jsou uloženy pro rychlé probuzení do aktivního režimu. Obsah paměti RAM se ukládá v závislosti na typu mikroprocesoru. Procesor je možné probudit externím probuzením nebo interními zdroji, jako jsou analogový komparátor či počítadlo pulzů. Průměrná spotřeba je mezi 10 až 30 μA. Čas potřebný k návratu do aktivního stavu je 5 až 8 μs.
Vypnutý (Off)	EM4	Nejnižší spotřeba, avšak na úkor nejdelšího času pro přechod do aktivního stavu. Vypnuty všechny hodinové signály a funkce procesoru s výjimkou monitorování přerušení. Obsah paměti a registrů se ztratí. Průměrná spotřeba je 1,5 μA. Čas potřebný k návratu do aktivního stavu je 160 μs.

2.4 Popis rozhraní pro přenos dat

Každé Arduino disponuje několika komunikačními rozhraními na HW úrovni. Mezi nejčastěji používané standardy patří I2C (Inter-Integrated Circuit) a SPI (Serial Peripheral Interface). Byly vytvořeny tak, aby poskytovaly jednoduché způsoby přenosu digitálních informací mezi senzory a mikrokontrolery, jako je Arduino. Knihovny Arduino pro I2C i SPI usnadní používání obou těchto protokolů. Volba mezi I2C a SPI je obvykle určena moduly, které je chceme připojit. Některá zařízení poskytují oba standardy, ale obvykle podporují jeden nebo druhý.

Sběrnice SPI (Serial Peripheral Interface) představuje jednu z forem sériových externích sběrnic. Tento vysokorychlostní komunikační protokol slouží pro vzájemné propojení dvou či více komunikujících uzlů, přičemž jeden uzel obvykle vystupuje v roli takzvaného řadiče sběrnice (master), ostatní uzly pracují v režimu slave. V našem případě je master Arduino a slave modul RC522. Uzel, který pracuje jako master, obsahuje generátor hodinového signálu, který je rozveden do všech ostatních uzlů, čímž je umožněn zcela synchronní (navíc ještě obousměrný) přenos dat. Master dále určuje, se kterým zařízením bude komunikovat. Sběrnice se používá pro komunikaci mezi řídícími mikroprocesory a ostatními integrovanými obvody (EEPROM, A/D převodníky, displeje...). Adresace se provádí pomoci zvláštních vodičů, které při logické nule aktivují příjem a vysílání zvoleného zařízení.

Jako první s tímto protokolem přišla firma Motorola a dnes již velká řada výrobců procesorů nabízí obvody s tímto rozhraním. Tyto obvody mají integrované řadiče rozhraní SPI na čipu procesoru, což umožňuje realizovat snadnější a hlavně mnohem rychlejší komunikaci s periferním obvodem. Přenos dat po sběrnici SPI je povinně obousměrný a sběrnici tvoří tři vodiče. Na začátku komunikace Master nastaví log. 0 na SS zařízení, se kterým chce komunikovat. Následně generuje hodinový signál na SCK, který zajišťuje synchronizaci. Data jsou přenášena signály MISO (Master In Slave Out) - přenos z periférie do mikroprocesoru a MOSI (Master Out Slave In) – přenos z mikroprocesoru do periférie mezi osmi- nebo šestnáctibitovými posuvnými registry řadiče a podřízeného obvodu. [19], [20]

Protokol I2C - Inter-Integrated Circuit je sběrnice vynalezená společností Philips. Sběrnice I2C je tvořena dvojicí vodičů, na kterých je v klidovém stavu díky Pull-up rezistorům stav logické 1. Vodič označený jako SCL generuje hodinový synchronizační signál pouze na straně Master (Arduino) a vodič označený jako SDA slouží k obousměrnému synchronizovanému přenosu dat. Jednotlivé fyzické vývody (SCL, SDA) těchto obvodů jsou vnitřně vybaveny budiči s otevřeným kolektorem, které umožňují připojení více obvodů na tuto sběrnici, aniž by došlo ke kolizím. Dále umožňují kterémukoliv obvodu kontrolovat bitovou synchronizaci a přizpůsobit tak rychlost přenosu dat. Přenosová rychlost je tedy přizpůsobena nejpomalejšímu modulu na sběrnici. [19], [21]

3 Návrh a realizace systému pro identifikaci závodníků

Copak je to za vojáka je skautská akční hra na motivy stejnojmenného filmu. Úkolem účastníků je dostat ze startu do cíle v co nejkratším čase. Herní území se rozkládá na ploše zhruba 20 km^2. Účastníci se mohou libovolně po tomto území pohybovat. Na začátku má každý účastník čtyři životy, o které může přijít, když narazí na vojenskou policii, která mu z ruky strhne pásku. Za každý ztracený život se ke celkovému času přičte 45 minut. Na předem známých místech účastníky čekají kontrolní stanoviště, na která však nemusí dojít, ale přijdou tak o možnost získat časový bonus. Ukázka mapy herního území je v příloze na obrázku A. Vítězem se stává účastník s nejkratším časem stráveným na trati. Stanoviště na trase jsou rozdělena do dvou kategorií - probíhací a úkolová.

Probíhací stanoviště - při doběhnutí na místo, účastník přiloží čip k zařízení a následně může pokračovat dál. Na těchto stanovištích se na čip zapíše číslo stanice a aktuální čas. Při závěrečném vyhodnocení se v takovém případě účastníkovi odečte 20 minut z celkového času. Tato stanoviště jsou rozmístěna po trase ve skupinách po třech. Na mapě (obrázek A) je jedna skupina označena např. „RUN 1a, RUN 1b, RUN 1c“. Pro účastníky to znamená, že si mohou vybrat jen jedno stanoviště ze skupiny, na které poběží. Pokud doběhnou na dvě stanoviště z jedné skupiny, započítá se jim jen 1 časový bonus.

Dále jsou na trase **úkolová stanoviště** - na těchto stanovištích budou dvě zařízení. Při doběhnutí na toto stanoviště přiloží čip k prvnímu zařízení. Dále má na výběr. Buď poběží dál, tudíž se mu odečte jen 20 minut stejně jako u probíhajícího stanoviště, nebo splní úkol, který je na daném místě připraven (střelba na terče, zdolat horolezeckou stěnu...). Za splnění úkolu může získat až 30 bodů = 30 minut časový bonus. Před opuštěním stanoviště účastník přiloží čip k druhému zařízení, které opět zapíše aktuální čas. Čas, který stráví na stanovišti, se tedy nezapočítává do výsledného času.

Pro lepší přehlednost uvádím tabulku bodování.

Tabulka 3.4: Přehled bodování

Kde	Co	Změna celkového času
probíhací stanoviště	příchod 2. ve stejné skupině	- 20 minut 0
úkolová stanoviště	příchod úkol čekací čas	- 20 minut až - 30 minut - čas na stanovišti
hrací pole	ztráta života	+ 45 minut

3.1 Požadavky na systém

Systém se bude skládat ze dvou hlavních částí. Jednak ze zařízení ukrytých v lese, která budou zapisovat data na čipy, a dále ze softwaru, který přečte data z čipů a spočítá výsledky.

Požadavky na zařízení:

- Pomocí RFID zapíše data na čip,
- ukryté ve vodotěsném obalu,
- každé zařízení má své unikátní číslo,
- obsahuje modul reálného času, který zná aktuální čas,
- dá účastníkům dostatečnou zpětnou vazbu o provedení zápisu.

Požadavky na vyhodnocovací systém, který se skládá z jednoho zařízení obsahující RFID anténu pro čtení dat z čipů, konektor pro připojení k PC, a dále z programu vytvořeném v databázovém systému MS Access:

- Vypočte celkový čas na trati (cíl - start),
- odečte 20 minut za návštěvu probíhajícího stanoviště,
- odečte čas strávený na úkolovém stanovišti,
- odečte čas přepočítaný z bodů na úkolovém stanovišti (maximálně 30 bodů, 1 bod = 1 minuta),
- přičte čas za ztracené životy (1 ztracený život = 45 minut),
- počet ztracených životů a body za úkoly se budou zapisovat ručně do tabulky.

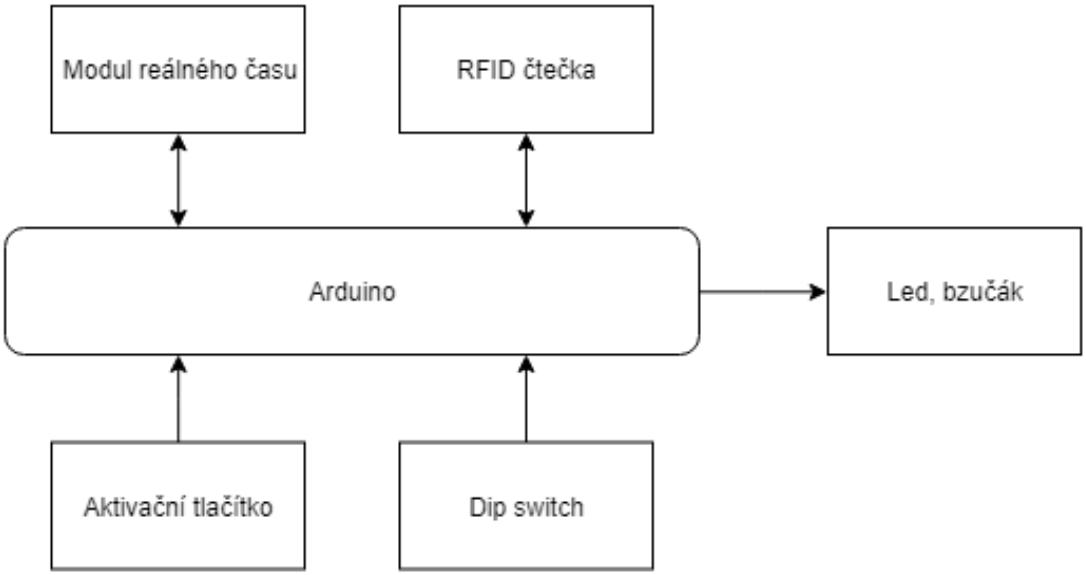
3.2 Hardwarová realizace zařízení

Tato část popisuje jednotlivá zařízení, která byla použita pro vytvoření systému. Jedná se o popis platformy Arduino Uno, RFID čtečky, modul reálného času a dalších součástek.

Jak je vidět na obrázku 3.7, v našem případě je Arduino hlavní částí zařízení. Po stisknutí aktivačního tlačítka se přivede napájení na RFID anténu. Když čtečka zjistí přítomnost čipu, načte z modulu reálného času aktuální čas a z DIP (Dual In-Line Package) přepínače zjistí ID stanice. Tyto údaje následně pošle na RFID čtečku, která je zapíše na čip. Pro navrhovaný prototyp je pro napájení zařízení použita 9V baterie, připojená přes sousosý konektor.

3.2.1 Arduino UNO

Základním stavebním prvkem systému je Arduino UNO. Je to základní programovatelná deska s 8bitovým mikrokontrolerem Atmega328P. Obsahuje 2 KB statické paměti RAM (Random-Access-Memory) a 1 KB paměti EEPROM (Electrically Erasable Programmable Read-Only Memory). Na Arduino desce se nachází 14 digitálních vstupních / výstupních pinů. V těchto 14 pinech 6 pinů může být také použito jako pulzní šířka modulačních pinů a 6 analogových I / O pinů. Dále je na desce napájecí konektor a resetovací tlačítko.



Obrázek 3.7: Diagram zařízení

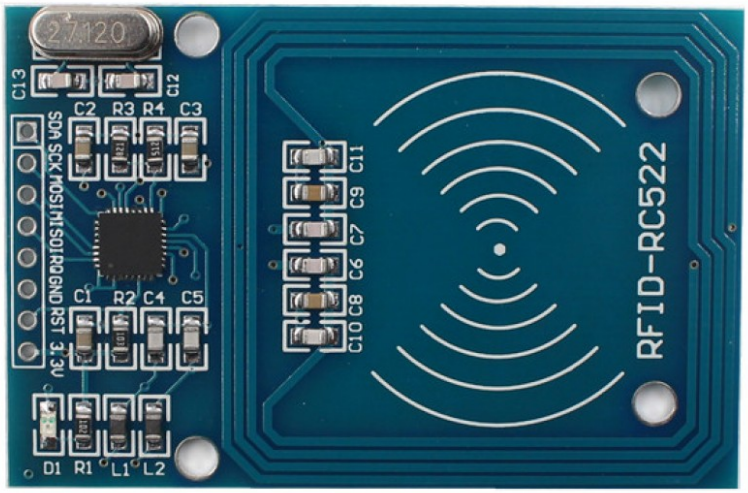
3.2.2 RFID čtečka

Nejprve čtečka vyšle impuls o délce asi 50 ms. Tento impuls je přijat anténou na čipu, která musí být naladěna na stejnou frekvenci jako anténa čtečky. Pomocí přijaté energie je nabit interní kondenzátor. Následně čip okamžitě vyšle svá data zpátky ke čtečce. Jako energie pro vysílání čipu slouží právě napětí, které je uchovano v kondenzátoru. Délka přenášených dat je 128 bitů i s zabezpečovacím kódem a přenos těchto dat trvá přibližně 20 ms. Data jsou zachycena anténou čtečky a dekódována. Perioda mezi dvěma cykly je mezi 20 ms až 50 ms. [22]

Popis: Modul RFID čtečky pro mikrokontrolery s čipem a kartou, typ MF RC522, frekvence 13.56 MHz, NXP, detekce chyb v přenosu ISO14443A, podpora šifrování CRYPTO1, obousměrná rychlost přenosu dat až 424 kbit/s, SPI komunikační sběrnice a integrovaná anténa.

Provozní proud: 13 až 26 mA / 3.3 V DC, klidový proud: 10 - 13 mA / 3.3 V DC, provozní teplota: -20 až 80 °C, napájení: 3.3 V DC, rozměry: 40 x 60 mm. [23]

Pro propojení RFID čtečky s Arduino deskou je zapotřebí propojit celkem 7 vodičů. Propojíme SDA s pinem 10, SCK s pinem 13, MOSI s pinem 11, MISO s pinem 12, GND se zemí, RST s pinem 9 a 3.3V s pinem 3V3 na Arduinu. Modul čtečky komunikuje s Arduinem pomocí SPI protokolu, a proto je nutné využít tyto vybrané piny na Arduino desce. Je možné změnit pouze piny SDA a RST, přičemž jejich změnu je nutné provést také na začátku programu.



Obrázek 3.8: RC522 čtečka

Paměť RFID čipu

Jak už je uvedeno výše, jsou různé typy RFID čipů. Z některých čipů je možné jen číst, další nám umožní jednorázový zápis a následné neomezené čtení a na třetí skupinu čipů je možné opakovaně zapisovat. Našim potřebám vyhovuje jen ten poslední typ s možností zápisu, konkrétně je použitý čip MIFARE Classic 1K Memory Layout.

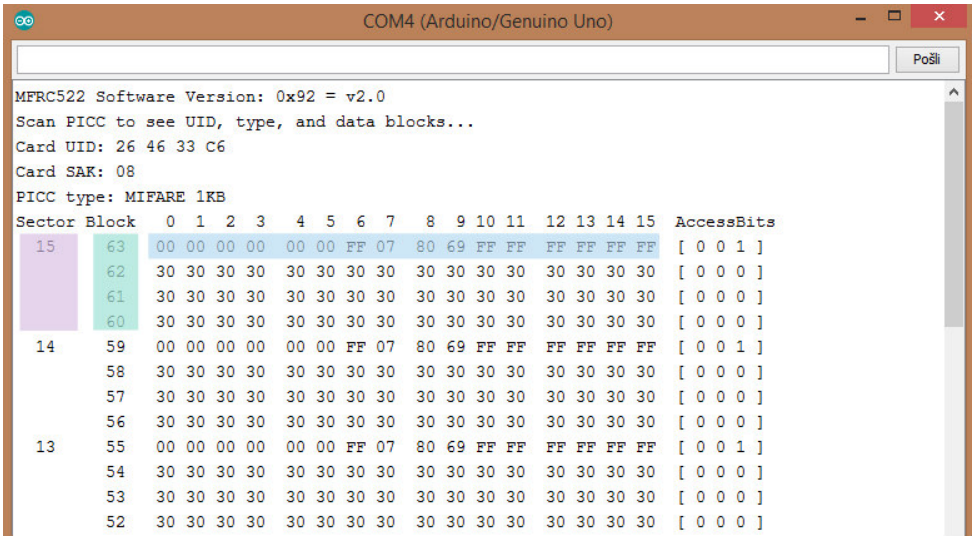
Vysvětlení jednotlivých pojmů: **MIFARE** je registrovaná ochranná známka NXP Semiconductors ze série čipů, které se běžně používají v bezkontaktních čipových kartách. **MIFARE Classic** používá vlastní protokol, který je v souladu s částmi 1-3 normy ISO / IEC 14443 typ A, s protokolem zabezpečení NXP pro autentizaci a šifrování. **1K Memory** - paměť o velikosti 1024 B.

1K paměť čipu je organizována v 16 sektorech (od 0 do 15).
Každý sektor je dále rozdělen na 4 bloky (blok 0 až 3).
Každý blok obsahuje 16 bajtů dat (od 0 do 15).

Ve výsledku máme tedy:

16 sektorů x 4 bloků x 16 bytů dat = 1024 bytů = 1K paměti

Paměť 1K se sektory, bloky a daty je zvýrazněna níže.



Obrázek 3.9: Reprezentace rozložení dat MIFARE Classic 1K

I když je uvedena velikost paměti 1 024 B, není možné ji celou využít, z důvodu zabezpečení přístupu k datům. Blok 3 každého sektoru se nazývá Sector Trailer a obsahuje informace nazvané Access Bits, které umožňují přístup (číst a psát) ke zbývajícím blokům v sektoru. To znamená, že pro ukládání dat jsou skutečně k dispozici pouze 3 dolní bloky (blok 0, 1 a 2), z toho plyne, že pro vlastní potřebu máme k dispozici pouze 48 bajtů z 64. Také blok 0 sektoru 0 je známý jako Manufacturer Block obsahující údaje o výrobci (IC) a jedinečný identifikační identifikátor (UID). [24]

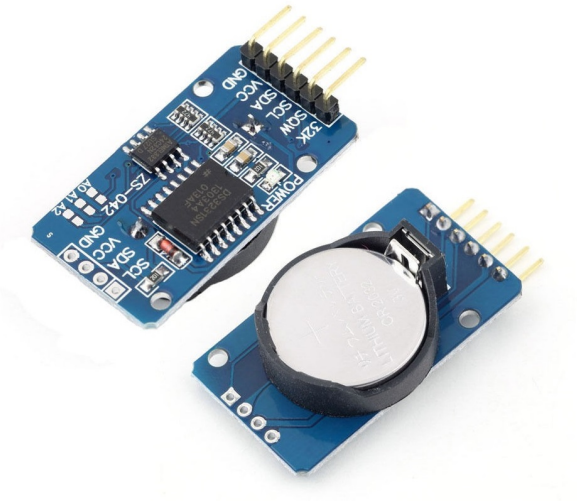
3.2.3 Modul reálného času

Modul DS3231 uchovává informace o aktuálním čase a datu. Podporuje časové údaje jako: sekundy, minuty, hodiny, dny, měsíce i roky. Mezi užitečné funkce patří: korekce pro přestupný rok, datum ukončení, dva nastavitelné alarmy a kalendář, přepínání režimů AM a PM. Všechny tyto záležitosti kolem času bychom mohli provádět i přímo výpočty v Arduinu, protože obsahuje oscilátor a časovač. Nedosáhli bychom ale takové variability použití a hlavně přesnosti jako s reálným časem RTC.

Modul obsahuje dva obvody DS3231 a AT24C32. První obvod DS3231 je označovan jako RTC – Hodiny reálného času. A druhý obvod AT24C32 je paměťový modul typu EEPROM.

Obvod RTC DS3231 vyniká oproti svým konkurentům vysokou přesností. Té je dosaženo integrovaným teplotně kompenzovaným krystalem, který má v rozsahu teplot 0 - 40 °C přesnost 2 ppm. Komunikace s celým modulem probíhá po sběrnici I2C a napájecí napětí je možné použít 3,3 nebo 5 V. Pro paměťový obvod pak také můžeme nastavit až osm různých adres díky propojkám A0-A2 na modulu. A pro zálohování uložených informací v obvodech při odpojení napájení, můžeme využít slot na baterii typu 2032.

Pro potřeby našeho zařízení použijeme tyto údaje:



Obrázek 3.10: RTC - Hodiny reálného času

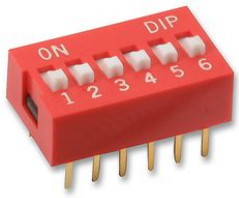
- Sekundy,
- minuty,
- hodiny,
- den.

Pro úspěšné propojení modulu DS3231 a Arduino desky stačí zapojit celkem 4 vodiče. Propojíme VCC s pinem 5V Arduina a GND se zemí Arduina. Pro piny SDA a SCL musíme vybrat vždy vyhrazené I2C piny na námi vybrané desce.

3.2.4 Vícenásobný přepínač

DIP switch je vícenásobný přepínač zapájený přímo do desky plošných spojů (PCB). Tvoří ho několik dvojic kontaktů, které lze nezávisle na sobě spojit nebo rozpojit. Jednotlivé kontakty jsou uspořádány ve dvou řadách a zapouzdřeny v keramickém či plastovém pouzdře - tzv. Dual In-Line Package (DIP). Celému přepínači se pak říká DIP switch. Rozpojení či spojení každé konkrétní dvojice kontaktů je přitom určeno polohou malé páčky, která se posouvá. Dvě význačné polohy každé páčky jsou označovány jako ON (zapnuto, což odpovídá propojení příslušných kontaktů), resp. OFF (vypnuto, rozpojení kontaktů).

Při konstrukci zařízení je tento modul využit pro určení ID stanice. Použijeme šestimístný DIP switch. To znamená, že máme 6 přepínačů, které mohou být ve stavu buď log. 0 nebo log 1. Tudíž můžeme vytvořit $2^6 = 64$ možností. Nejednoduší způsob, jak tento modul zapojit, je, že každá pozice spínače je připojena k samostatnému digitálním pinu Arduino. Pozice, která je „vypnutá“, je uzemněna.



Obrázek 3.11: DIP switch

3.2.5 Ostatní součástky

Mezi další součástky, které jsou použity pro tvorbu zařízení, patří tlačítko, jako vstupní zařízení, a dále LED (Light-Emitting Diode), neboli světlo emitující dioda, nebo piezo bzučák jako výstupní zařízení.

Tlačítko
Hlavní funkcí tlačítka je přepnout Arduino z režimu spánku do aktivního režimu. Následně se aktivuje RFID čtečka a bude umožněna komunikace mezi čipem a zařízením. Toto tlačítko bude sloužit hlavně pro uživatele - účastníky závodu, takže je potřeba ho umístit tak, aby k němu byl umožněn dostatečný přístup i v případě, že naše zařízení je schovano ve vodotěsné krabičce. Více o ovládání zařízení je v kapitole 3.3.6.

LED
Light-Emitting Diode - světlo emitující dioda, to znamená, že pokud touto součástkou projde elektrický proud v propustném směru, tak dioda díky P-N přechodu začne vyzařovat světlo.

Bzučák
Označení „piezo“ nesou součástky, které pro svou činnost využívají piezoelektrický jev. „*Je to schopnost krystalu generovat elektrické napětí při jeho deformování, nebo také jev opačný, kdy je na krystal přiloženo napětí a on se tvarově deformuje.*“[25]

Obrázek 3.12: LED a piezo bzučák

V našem případě nás zajímá jev opačný, neboli nepřímý piezoelektrický jev. Přivedeme-li na krystal napětí, dochází k jeho deformaci. Přivedením měnícího se napětí na krystal dochází

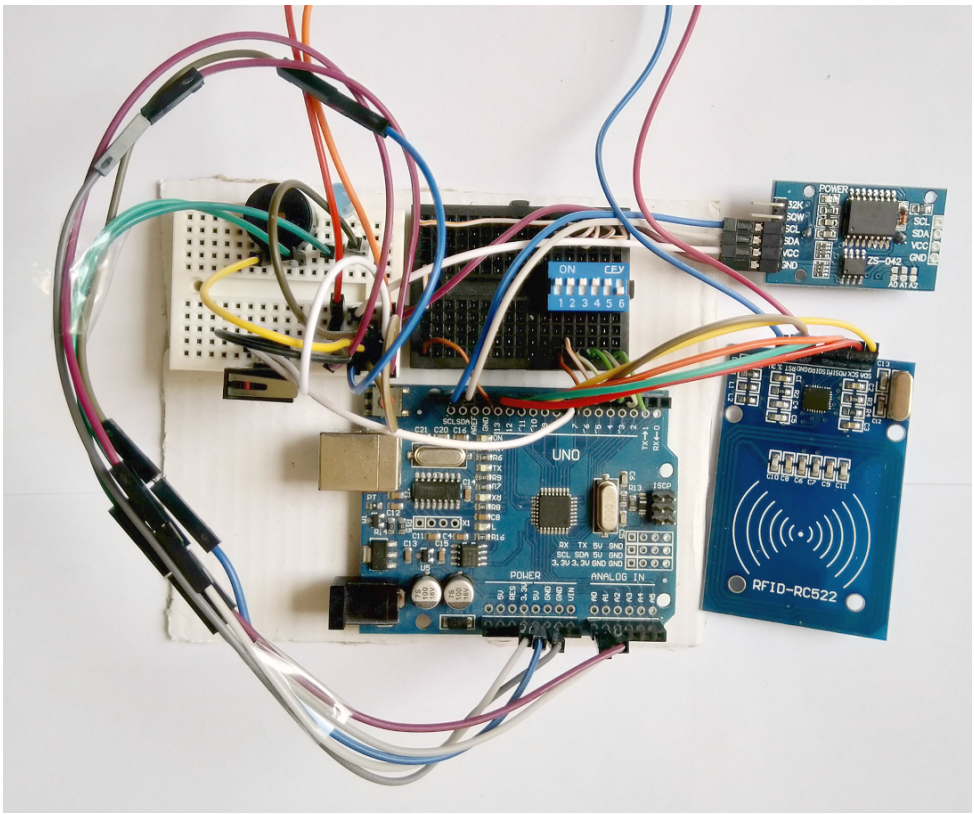
k jeho střídavému smršťování a rozpínání. Díky pohybu krystalu se rozpohybují i okolní částice, a vzniká tak zvuková vlna, kterou slyšíme. [26]

To znamená, že když přivedeme na bzučák napětí, ozve se jen lupnutí. Pro naše potřeby je ale nutné, aby bzučák vydával nějaký tón po dobu např. 2 sekund. Z toho plyne, že musíme pin, ke kterému je bzučák připojen, rychle vypínat a zapínat.

Bzučák a LED diody budou sloužit jako ukazatel pro uživatele, že zařízení provedlo požadovaný úkol, příkladem může být správný zápis dat na čip. Více o jejich funkci je rozebráno v kapitole 3.3.6.

Pro správnou funkci tlačítka potřebujeme zapojit tři drátky - napájení (5V), GND a 1 digitální vstup. Z důvodů, že toto tlačítko bude využito pro přerušení, je potřeba digitální výstup zapojit na pin 2. Pro zapojení LED potřebujeme dva piny. Jeden jako datový vstup - anoda a druhým se komponenta uzemní - katoda. U bzučáku je podobná situace, také potřebujeme připojit jeden vstupní pin a GND. Tentokrát použijeme pin A1.

Na závěr této kapitoly, ve které jsou vypsány použité komponenty, je uvedena fotografie reálného zapojení navrhovaného řešení. V příloze B je uvedeno jeho schéma zapojení.



Obrázek 3.13: Fotografie zapojení navrženého řešení

3.3 Softwarová realizace zařízení

V této kapitole se zaměříme na software, který je nahraný v zařízení. Postupně si popíšeme, jak funguje Arduino a jednotlivé funkce, od počátečního nastavení hodin, přes aktivaci Arduina, až po zapsání dat na čip.

3.3.1 Potřebné knihovny

Důležitou součástí při programování Arduina jsou knihovny. Jsou ve formě složek, které sdružují více souborů se zdrojovými kódy. Díky těmto zdrojovým kódům máme jednodušší a přehlednější vlastní program. Nejvíce používáme knihovny, když chceme připojit nějaký modul, který má složitější ovládání, jako je například displej, RFID čtečka a další. Vzhledem k tomu, že Arduino je open-source platfoma, je možné na internetu stáhnout velkou spoustu knihoven. V této práci jsou použity tyto knihovny:

- MFRC522 - RFID čtečka,
- DS3231 - modul hodin reálného času,
- SPI - SPI sběrnice,
- Wire - I2C sběrnice,
- Sleep - Přepnutí do režimu spánku.

3.3.2 Programování Arduina

Pro naprogramování aplikace bylo využito vývojové prostředí Arduino IDE. Toto prostředí obsahuje textový editor a tlačítka pro nahrání kódu do Arduina. Dále obsahuje nástroj serial monitor, který umožňuje komunikaci s Arduino deskou. Na straně hardwarové části pro tyto účely slouží knihovna Serial.

Každý program pro Arduino obsahuje dvě funkce. Jsou pojmenované `setup()` a `loop()`. Kód ve funkci `setup()` se provede vždy po zapnutí nebo restartu zařízení. Patří zde hlavně inicializace proměnných. Tyto příkazy se provedou pouze jednou. Následuje smyčka `loop()`, do které patří kód, který se bude vykonávat do nekonečna, tedy do vypnutí zařízení.

V našem případě jsou na začátku kódu uvedeny použité knihovny, které jsou vypsany výše. Dále následuje deklarace proměnných, inicializace použitých modulů a komunikačních sběrnic. Je potřeba uvést, na kterých pinech je připojen DIP switch, nebo na kterém pinu je připojen reset RFID čtečky. Dále se nachází smyčka `loop()`, ve které se volají jednotlivé funkce. Tyto funkce jsou popsány níže. To, jaká funkce se zavolá, se odvíjí od stavu DIP přepínače. Základní funkce použité pro zařízení jsou:

- **0** - Nastavení času,
- **10 - 63** - Zápis o průchodu stanovištěm. V tomto případě slouží číslo nastavené pomocí DIP přepínače jako ID stanice.

3.3.3 Nastavení aktuálního času v zařízení

Jedna z otázek je, jak nastavit aktuální čas ve všech zařízeních (30 ks). Modul hodin reálného času má záložní zdroj energie v podobě vlastní baterie. Ta může dodávat modulu energii, i když je hlavní napájení odpojeno. Takže poskytované informace modulem tedy mohou být vždy aktuální i při odpojení hlavní baterie. Ale vzhledem k tomu, že zařízení se budou používat jednou nebo dvakrát do roka, je opotřebení baterie, v čase, kdy se zařízení nebude používat, nevýhodné.

Jsou dvě možnosti, jak aktuální čas do zařízení zapsat. Buď zařízení připojíme k počítači a čas nahrajeme přes USB kabel, nebo čas přeneseme bezdrátově pomocí čipu. Aby se před použitím těchto zařízení nemuselo každé zařízení připojovat k počítači, byla vybrána druhá varianta. Při nastavování aktuálního času pomocí čipu sice bude potřeba zařízení, které bude připojené k počítači, ale toto jedno zařízení bude stačit k obsluze všech třiceti zařízení současně. Tahle varianta nejspíše nezaručí velkou přesnost, ale to pro potřeby našeho zařízení je dostačující.

Proces nastavení hodin lze rozdělit do dvou fází: zapsání času na čip a zapsání času do zařízení.

Z PC na čip

Pro tuto část je nezbytnost mít Arduino s RFID anténou propojené s počítačem. Dále je třeba mít na počítači spuštěnou aplikaci, která bude sloužit k obsluze Arduina. Prozatím nám pro tyto potřeby poslouží vývojové prostředí Arduino IDE. Po zvolení možnosti „Zápis hodin“ se po sériové lince vyšle signál pro Arduino, které poté na každý přiložený čip zapíše aktuální hodiny. Tato funkce se bude neustále opakovat. Toto opakování je z důvodu zachování alespoň částečné přesnosti času na zařízeních. To znamená, že po zapsání hodin na čip, je možné tento čas zapsat do maximálně pěti zařízení. Pokud chceme zapsat čas i do dalších zařízení, je potřeba na čip znovu zapsat aktuální čas z počítače.

Samotný zápis se provede pomocí funkce `MIFARE_Write()`, která je popsána níže (kapitola č. 3.3.5).

Z čipu na zařízení

Jak už je uvedeno výše, zařízení budou mít zapojený DIP switch. V případě, že na něm bude nastavená hodnota 0 (000000), bude zařízení čekat na čip, který bude mít v prvním bloku dat napsaný čas. Tento čas se pak pomocí funkce `rtc.setDateTime()` nastaví v modulu hodin reálného času.

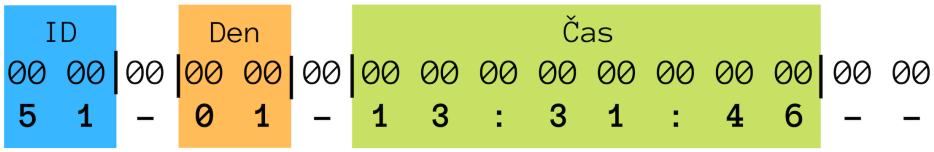
3.3.4 Zápis o průchodu stanovištěm

Jedná se o nejdůležitější funkci, kterou budou zařízení vykonávat.

Nejprve je potřeba projít zapsaná data na čipu a zjistit, zdali už záznam o průchodu tímto stanovištěm není na čipu zapsán, což znamená, že závodník toto stanoviště ještě nenavštívil. V případě, že tomu tak není, je dalším požadavkem najít volný blok dat a na něj zapsat data.

Mezi data, které jsou potřeba zapsat, patří: ID stanice, den v týdnu a čas. V budoucnu je možné přidat funkci, která na čip zapíše body, které účastník obdrží na úkolovém stanovišti. **ID stanice** je dvojciferné číslo, které je určeno pomocí hardwarové součástky DIP switch. Toto číslo může nabývat hodnot od 10 - 63. **Den v týdnu a aktuální čas** zjistíme pomocí modulu hodin reálného času.

Pro zapsání dat o průchodu jedním stanovištěm, použijeme jeden blok dat, tj. 16 B. Na obrázku č. 3.14 je zobrazen příklad ukládaných dat čip (ID stanice = 1, Den = 1, Čas = 13:31:46). Poslední dva bajty jsou nevyužity.



Obrázek 3.14: Ukázka dat na čipu

3.3.5 Komunikace mezi RFID anténou a čipem

Komunikace mezi čipem a zařízením nepatří mezi jednoduché úkoly. Naštěstí spoustu práce odvede knihovna MFRC522, která zjednodušuje čtení a zápis na čipy RFID.

Na začátku je potřeba načíst knihovnu MFRC522 a SPI, nadefinovat piny Arduina, na kterých je modul RC522 (RFID čtečka) připojen, a vytvoření objektu.

```
#include <SPI.h> // připojení knihovny pro komunikaci přes SPI sběrnici
#include <MFRC522.h> // připojení knihovny pro RFID čtečku

#define SS_PIN 10 // slave select pin
#define RST_PIN 5 // reset pin
MFRC522 mfrc522(SS_PIN, RST_PIN); // vytvoření instance MFRC522
MFRC522::MIFARE_Key key; // vytvořeni struktury MIFARE_Key s názvem key, která
    bude obsahovat informace o kartě
```

Ve funkci `setup()` se provede inicializace komunikace po SPI sběrnici a objektu MFRC522. Je také potřeba připravit bezpečnostní klíč. V tomto případě je všech šest bajtů nastaveno na

0xFF, protože nové RFID čipy nemají klíče definovány. Kdybychom měli čip, který již někdo naprogramoval, museli bychom tento klíč znát, abychom mohli k čipu přistupovat.

```
SPI.begin();      // Inicializace SPI sběrnice
mfrc522.PCD_Init();      // Init MFRC522
Serial.println("Scan a MIFARE Classic card");

// Příprava bezpečnostního kódu.
for (byte i = 0; i < 6; i++) {
   key.keyByte[i] = 0xFF; // keyByte je definován ve struktuře MIFARE_Key v
       knihovně MFRC522
}
```

Ve smyčce `loop()` se nacházejí dvě funkce, které navážou komunikaci s čipem. Aby tyto funkce splnily svůj účel, je nutné, aby se tyto dvě funkce nacházely na začátku funkce `loop()`. Jedná se o funkci `mfrc522.PICC_IsNewCardPresent()`, která vrátí jedničku, pokud se v okolí antény nachází RFID čip. Pokud se zde žádný čip nenachází, je pomocí příkazu `return` smyčka `loop()` ukončena. V případě úspěšné detekce čipu se zkontroluje jeho správné načtení funkcí `mfrc522.PICC_ReadCardSerial()` a ještě následuje kontrola, jestli načtený čip patří mezi podporované typy. Dále může následovat obsloužení již načteného čipu.

```
void loop() {
  // kontrola RFID čipů v okolí modulu,
  // pokud není žádný čip v okolí, volá se loop funkce od začátku,
  if ( ! mfrc522.PICC_IsNewCardPresent()) {
    return;
  }
    // kontrola správného přečtení RFID čipu
  if ( ! mfrc522.PICC_ReadCardSerial()) {
    return;
  }
  .....
```

Nyní máme úspěšně návaznou komunikaci mezi RFID čtečkou a čipem a je možné provést čtení nebo zápis na čip. Ještě je zapotřebí definovat proměnné pro tyto funkce. Mezi potřebné proměnné patří číslo bloku, na které budeme provádět zápis nebo čtení. V tomto případě je vybrán druhý blok z nultého sektoru. Nesmíme zapomenout, že na každý třetí blok v sektoru není možné zapisovat. Dále pro zápis potřebuje 16bajtové pole, do kterého vložíme text, který bude mít maximálně 16 znaků. V případě, kdy chceme blok smazat, vložíme do tohoto pole nuly. I pro potřeby čtení musíme definovat pole. V našem případě budeme číst 16 bajtů, takže potřebujeme pole pro 18 bajtů. Ty dva bajty navíc jsou z důvodu toho, že funkce `MIFARE_Read()` v knihovně

MFRC522 vyžaduje vyrovnávací paměť, která má alespoň 18 bajtů pro načtení 16bajtového bloku.

```
int block = 2; // číslo bloku, do kterého se bude zapisovat nebo číst
byte blockcontent[16] = {"-Toto-se-zapise-"}; // blok pro zápis
byte readbackblock[18]; // blok pro čtení
```

Pro zápis je vytvořená funkce `writeBlock(block, data)`. Má dva vstupní parametry. Prvním je číslo bloku, na který se má provést zápis, a druhým jsou zapisovaná data. Uvnitř funkce se provede kontrola, že na zadaný blok je možné zapisovat. To znamená, že to není třetí blok v sektoru. Následně pomocí funkce z knihovny MFRC522 `PCD_Authenticate()` se provede odemčení paměťového sektoru, ve kterém je daný blok, pomocí komunikačního protokolu Crypto-1. [27]. Pro samotný zápis je použita další funkce z knihovny MFRC522, a tou je `MIFARE_Write(block, data, size)`. Vstupní parametry do této funkce jsou číslo bloku, zapisovaná data a velikost datového bloku.

Funkce pro zápis má návratovou hodnotu, podle které můžeme zjistit, že zápis byl proveden v pořádku. I v takovém případě je vhodné správnost zápisu dále ověřit. Pro tyto účely můžeme použít funkci pro čtení dat z čipu. Přečteme právě zapsaný blok a tato přečtená data porovnáme s těmi, co jsme chtěli zapsat. Pro čtení je vytvořená funkce `readBlock(block, data, size)`, která má opět dva vstupní parametry. Těmi jsou číslo bloku a pole bajtů pro ukládání obsahu bloku. Uvnitř je opět odemčení daného sektoru a zavolání funkce `MIFARE_Read()`.

Ještě stojí za zmínku funkce `PICC_DumpToSerial()`. Tato funkce zobrazí celou paměť čipu.

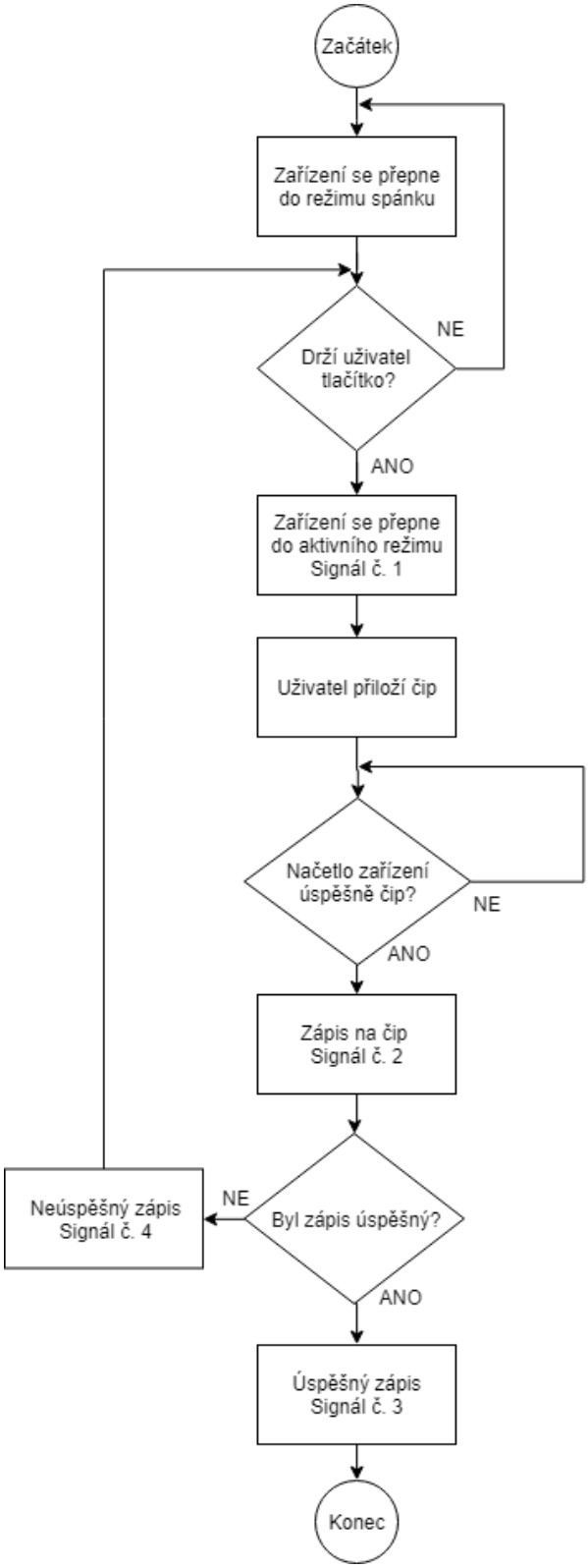
3.3.6 Ovládání zařízení

Když účastník, jenž doběhne na stanoviště, musí najít zařízení. Bude upevněno na tyčce a umístěno na viditelném místě. Toto zařízení obsahuje tlačítko. Po jeho stisknutí je možno přiložit k zařízení čip. Následně se provede zápis o průchodu stanovištěm. V případě, že vše proběhne správně, zazní jiná signalizace, než když nastane nějaká chyba.

V době, kdy na stanovišti nikdo nebude, bude Arduino přepnuto do režimu spánku (viz následující kapitola 3.3.7). Přerušení pro přepnutí do aktivního režimu se provede pomoci tlačítka. Dále je nezbytné, aby zařízení v dostatečné míře oznamovalo, co se aktuálně děje. V našem případě zařízení nemá žádný displej, takže signalizace bude probíhat jen za použití ledek a piezo bzučáku. Na obrázku č. 3.15 je pomoci vývojového diagramu znázorněno, jak zařízení funguje.

Po stisku tlačítka se Arduino přepne do aktivního režimu a zahájí příjem RFID antény. Následuje provedení **Signálu č. 1**. Ozve se krátký tón pomoci bzučáku a na okamžik se rozsvítí LED dioda.

V dalším kroku provede uživatel přiložení čipu. Po úspěšném načtení čipu se realizuje **Signál č. 2**. Opět zazní krátký tón a rozsvítí se dioda. V tomhle případě ale bude svítit po celou dobu následujícího zápisu.



Obrázek 3.15: Vývojový diagram ovládání

Zápis buď proběhne úspěšně, nebo nastane nějaká chyba. Při správném zápisu se provede **Signál č. 3**. Zazní další krátký tón a dioda, která je doposud rozsvícená, zhasne a na chvíli ještě blikne. Pak zařízení přejde do režimu spánku. Při neúspěšném zápisu se vykoná **Signál č. 4**. Zazní krátký zvuk zabliká LED dioda a zařízená se snaží znova navázat komunikaci s čipem.

Může se zdát, že všechny signály vypadají skoro stejně, ale opak je pravdou. Je potřeba tyto signály co nejvíce rozlišit. Zejména signály č. 3 a 4, které uživateli oznamují, jestli byl zápis proveden správně nebo ne. Ze bzučáku je možné například pouštět tóny o různé frekvenci. Pro rozeznění bzučáku je možné použít funkci `tone(pin, frekvence, dobaTrvani)`. Tato funkce má jako vstupní parametry pin, na který je bzučák připojen, frekvenci výsledného tónu a délku trvání v milisekundách.

3.3.7 Snížení spotřeby

Důležitým faktorem při otázce snížení spotřeby energie je, jak často bude zařízení vykonávat nějakou činnost. Jinak řečeno, jak často nějaký účastník doběhne na stanoviště a bude potřebovat zapsat záznam na čip. Z poznatků z předchozích ročníků můžeme určit, že:

- Akce se koná jednou ročně.
- Závod probíhá v sobotu od 8:00 do 16:00.
- Většina zařízení je roznesena už v pátek večer.
- Doba jednoho zápisu trvá průměrně 5 s.
- Účastníků je okolo 100.

Z výše uvedených informací si lze odvodit, že pohotovostní doba zařízení je okolo 24 hodin, zatímco doba, po kterou bude provádět nějaké instrukce, je 5 * 100 = 500 s = 8,3 minut. Z toho plyne, že je velice nevýhodné, aby zařízení běželo celou dobu na plný výkon. Pro zajištění co nejmenší spotřeby v době, kdy se zařízení nepoužívá, je vhodné přepnout Arduino do režimu spánku.

Když se u Arduina přepne režim do režimu spánku, tak Arduino vypne všechny nepotřebné komponenty, a sníží se tak spotřeba energie MCU (Microcontroller Unit). Jak je uvedeno výše, Arduino má více režimů spánku. Ten nejvíce účinný, to znamená, že po jeho zapnutí má Arduino nejmenší spotřebu, je `SLEEP_MODE_PWR_DOWN`. V tomto režimu je jediný způsob, jak jej můžeme probudit, a to je pomocí přerušení.

Nyní si popíšeme, co je potřeba pro uspání Arduina. Jako první je potřeba uvést, že piny, na kterých je možné nastavit přerušení, jsou pro Arduino Uno digitální piny 2 a 3. Tlačítko při stlačení spojuje dva body v obvodu. Pokud je ale tlačítko otevřené (nestlačené), tak mezi těmito body neexistuje žádné spojení a na pinu 2 je neurčitá hodnota. Jelikož interní pull-up rezistor na pinu 2 nebo 3 je aktivní a připojen k 5 voltům, při otevření tlačítka čteme HIGH. Když je tlačítko zavřené, spojení se zemí je úplné a Arduino jej čte jako LOW.

Na začátku je potřeba načíst knihovnu, která obsahuje vše, co je potřeba pro uspání Arduina. Dále definujeme proměnnou s číslem pinu, na kterém je připojené tlačítko, pomocí kterého přerušíme režim spánku Arduina.

```
#include <avr/sleep.h> // AVR knihovna
#define interruptPin 2 // pin, pomocí kterého se Arduino přepne zpátky do
    aktivního režimu
```

Následně ve funkci `setup()` je potřeba nastavit digitální pin 2 jako vstupní pin. Není však použité klasické `INPUT`, ale `INPUT_PULLUP`. Tímto způsobem se zajistí, že na pinu bude hodnota HIGH nebo LOW.

```
void setup() {
    pinMode(interruptPin, INPUT_PULLUP);
}
```

V hlavní funkci `loop()` nejdříve povolíme úsporný režim pomocí funkce `sleep_enable()`, jenž najdeme v knihovně avr/sleep. Bez zavolání téhle funkce by se Arduino nedalo přepnout do režimu spánku.

Jako další je funkce `attachInterrupt(přerušení, funkce, mod)`, pomocí které se aktivuje přerušení. Mezi její vstupní parametry patří přerušení. Je potřeba pomocí funkce `digitalPinToInterrupt(pin)` převést pin na přerušení. Arduino pro digitální pin 2 používá přerušení 0 a pro pin 3 přerušení 1. Dalším parametrem je název funkce, která se provede při odchycení přerušení. V našem případě je to funkce `wakeUp()`. Třetím vstupním parametrem je režim, který popisuje, kdy se přerušení spustí. Je na výběr ze 4 možností: [28]

LOW přerušení nastane vždy, když je pin v logické nule,

CHANGE přerušení nastane při změně logické hodnoty,

RISING přerušení s příchodem vzestupné hrany,

FALLING přerušení s příchodem sestupné hrany.

Dále nastavíme mód režimu spánku. V našem případě je použit ten nejvíce úsporný. Samotná změna režimu se provede pomocí funkce `sleep_cpu()`. Obě tyto funkce se nacházejí v knihovně avr/sleep. [29]

```
void loop() {
    sleep_enable(); //povolení přepnutí do režimu spánku
    attachInterrupt(digitalPinToInterrupt(interruptPin), wakeUp, CHANGE); //
        nastaví přerušení na pinu 2, spustí funkci wakeUp při změně hodnoty
    set_sleep_mode(SLEEP_MODE_PWR_DOWN); // volba úsporného režimu
    sleep_cpu(); // aktivace režimu spánku
```

```
    Serial.println("Active!"); // vypíše se až po probuzení
}
```

Při změně hodnoty na pinu 2 se provede funkce `wakeUp()` a pokračuje se ve funkci `loop()` od místa, kde došlo k přerušení. Ve funkci `wakeUp()` je vhodné zakázat přepínaní režimu a odpojit funkci od přerušení.

```
void wakeUp(){
    sleep_disable();
    detachInterrupt(0);
}
```

Jedna z dalších věcí, jak ušetřit spoustu energie, je v konečném zařízení použít Arduino Mini Pro. Arduino Mini Pro a Arduino Uno má velmi podobné vlastnosti, ale Mini má mnohem méně hardwaru, který potřebuje napájení, takže používá mnohem méně energie.

Pro příklad, Uno používá mezi 30-40 mA v aktivním režimu a asi 19 mA, když je v režimu spánku. Mini za běhu potřebuje sice 25 mA, ale po uspání jen 0,57 mA. Více o této problematice je popsáno v kapitole 3.5.3

3.4 Aplikace na vyhodnocení dat

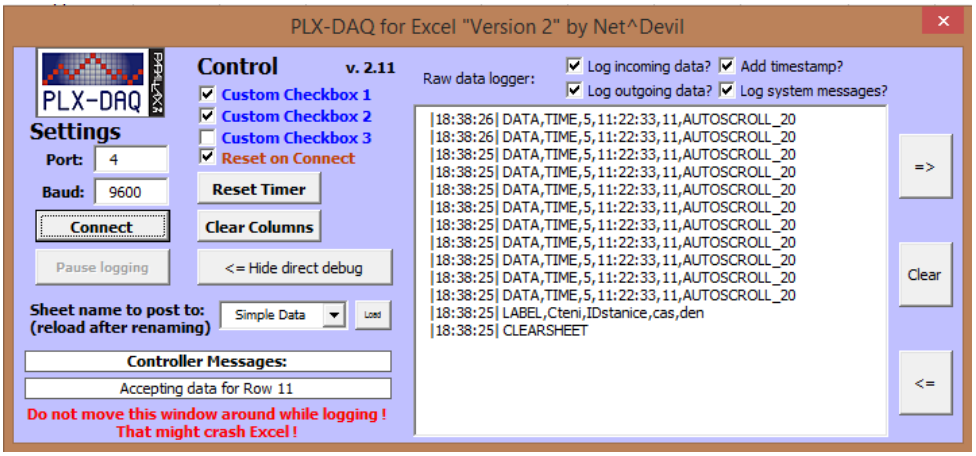
Tato část je rozdělena do 2 dvou částí. Pro nahrání dat ze zařízení do počítače je využitý doplněk pro Excel. Pro následné vyhodnocení výsledků je použita aplikace MS Access s kombinací s MS Excelem.

3.4.1 Export dat do Excelu

Pro tento úkol byl vybrán PLX-DAQ (Parallax Data Acquisition Tool). Je to formulář napsaný ve Visual Basicu. Tento program slouží k navázání snadné komunikace mezi Microsoft Excelem na Windows a jakýmkoliv zařízením, které podporuje protokol sériového portu. O vývoj se postarala firma Parallax Inc, která však v roce 2014 zveřejnila první a poslední verzi tohoto software. Na svých stránkách [30] uvádí mezi požadavky na systém - Microsoft Windows 98 a Excel 2000 až 2003, a upozorňují, že novější verze nejsou podporovány. Naštěstí na internetu lze najít i novou verzi, kterou je možné spustit na moderních systémech. Tu vytvořil NetDevil, člen fóra Arduino. Na obrázku 3.16 je snímek aktuální verze 2.11.

Nejdůležitější je levý sloupec aplikace. Je potřeba zde správně nastavit port, na kterém je Arduino připojeno, a rychlost přenosu dat v bitech za sekundu (baud) pro sériový přenos dat. Dále už stačí stitknout tlačítko „Connect“ a data se budou postupně vkládat do buněk v excelu.

Ještě je potřeba, aby zařízení posílalo data ve správném formátu. K těmto účelům se používá příkaz `Serial.println()`. Jednotlivé parametry jsou odděleny čárkou. Aplikace má spoustu vylepšení, lze například z Arduina poslat příkaz a v počítači se ozve pípnutí. Pro naši práci využijeme jen základní posílání dat.



Obrázek 3.16: Software PLX-DAQ

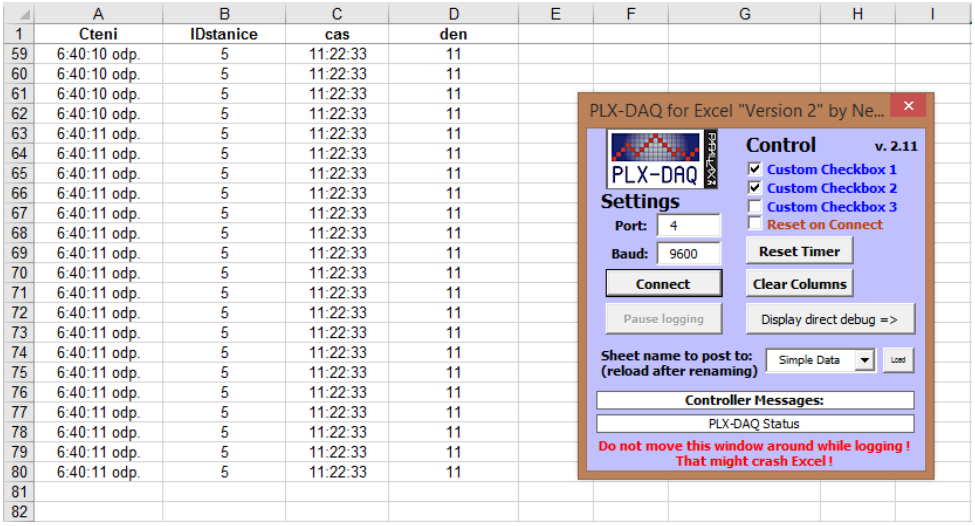
Na začátku nastavíme rychlost komunikace. Je nutné uvést stejné číslo jako ve formuláři v počítači. Dále je možné poslat příkaz na vymazaní dosavadních dat v excelu. Je možné smazat jen data, anebo i záhlaví sloupců. Pro nové pojmenování sloupců slouží příkaz, která začíná pojmem „LABEL“

```
void setup() {
    Serial.begin(9600); // nastavení rychlosti komunikace
    Serial.println("CLEARSHEET"); // smazání dat
    Serial.println("LABEL,CasCteni,IDstanice,cas,den"); // pojmenování sloupců
}
```

Uvnitř cyklu `loop()` je už jen důležité nezaměnit pořadí hodnot. Posílaná data musí začít textem „Data“. PLX DAQ rozpozná vyhrazená kódová slova DATE, TIME a TIMER a v Excelu budou nahrazena hodnotami. Například TIME bude přepsáno na aktuální čas počítače. K řetězci můžete přidat klíčové slovo `AUTOSCROLL_20`. Tato informace umožní PLX DAQ automaticky posouvat okno aplikace Excel s každým novým řádkem přijatých dat. Číslo (např. 20) udává, kolik řádků by mělo být zobrazeno nad posledním řádkem, který je automaticky posouván.

```
void loop() {
 Serial.println( (String) "DATA,TIME," + ID + "," + cas + "," + den + ",
     AUTOSCROLL_20");
}
```

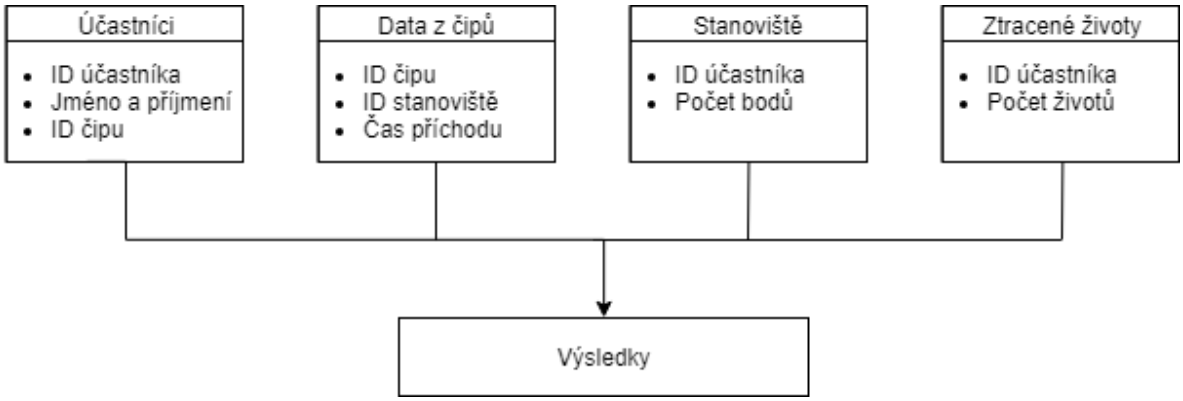
Na obrázku 3.17 jsou ukázána výsledná data v Excelu.



Obrázek 3.17: PLX-DAQ

3.4.2 MS Access

Nyní už jsou data z čipů v Excelu. Pro získání kompletních výsledků závodů je ještě potřeba tato data přepočítat, přidat bodování ze stanovišť a počet ztracených životů jednotlivých účastníků. Všechna tato data jsou zapsána v jednotlivých listech v MS Excel a následně v programu MS Access pomocí několika SQL (Structured Query Language) příkazů můžeme zjistit, kdo se umístil na pozici vítězů.



Obrázek 3.18: Schéma tabulek

Dvě hlavní věci, na které je potřeba se nejvíce zaměřit, je ověření jednotného zápisu ve skupině stanovišť a vypočtení času, který účastník strávil na úkolovém stanovišti.

Skupinou probíhacích stanovišť myslíme dvě až tři stanoviště, která jsou na mapě propojená čárou. Mapa herního území je uvedena v příloze A. Při návštěvě více stanovišť z jedné skupiny se započítá pouze jeden časový bonus. To ověří dotaz, který vypíše ID karty a sloupec Time naplní hodnotou 20, když nalezne záznam z alespoň jednoho stanoviště. Pro jednodušší počítání

výsledku jsou zařízení na stanovištích očíslovaná podle skupin, jak jsou rozdělena stanoviště v hracím poli. (RUN1a = 11, RUN1b = 12, RUN1c = 13)

Problematika úkolových stanovišť je řešena následujícím způsobem. Čas, který účastník stráví na stanovišti, se nezapočítává do celkového času. Pro tyto potřeby jsou na úkolovém stanovišti 2 zařízení. Jedním se zapisuje příchod na stanoviště a druhým odchod. Při vyhodnocení se rozdíl těchto dvou časů odečte od celkového času.

Na obrázku č. 3.19 jsou na ukázku upravené výsledky z minulého ročníku této hry. Je z nich možné vyčíst, komu při závodu nebyla ani jednou stržená páska. A pod čarou jsou uvedeni závodníci, který přišli o alespoň jeden život. Je možné si všimnout, že i když závodník s číslem 21 doběhl do cíle jako první, závod vyhrál závodník s číslem 43, který sice doběhl do cíle o hodinu a půl později, ale získal body na úkolovém stanovišti. Ve výsledku je jeho výsledný čas ještě o půl hodinu kratší.

	A	B	E	F	I	L	O	R	U	V	W	X	Y	Z	AA	AB	AC	AD
1	ID	Jmeno	Cil	CP1	CP2	CP3	CP4	CP5	RUN1	RUN2	RUN3	RUN4	RUN5	RUN6	Zivoty	Suma	CasNaTrati	VyslednyCas
2	43	Ondřej	12:01:35	80	0	0	0	0	0	20	0	0	20	20	0	140	4:01:35	1:41:35
3	3	Jan	11:48:50	0	0	0	0	0	20	20	20	20	20	0	0	100	3:48:50	2:08:50
4	21	Jaroslav	10:30:00	0	0	0	0	0	0	20	0	0	0	0	0	20	2:30:00	2:10:00
5	45	Lukáš	16:11:56	90	66	0	0	82	20	20	0	0	20	20	0	318	8:11:56	2:53:56
6	18	Jakub	12:39:34	0	0	0	20	0	20	20	0	20	20	0	0	100	4:39:34	2:59:34
7	28	Jan	12:39:38	0	0	0	20	0	20	20	0	20	20	0	0	100	4:39:38	2:59:38
8	56	Šimon	16:27:53	0	110	0	68	42	20	20	0	20	20	20	0	320	8:27:53	3:07:53
9	9	Jakub	12:39:36	0	0	0	20	0	0	20	0	20	20	0	0	80	4:39:36	3:19:36
10	14	Zdeňka	16:10:20	0	79	0	97	64	0	0	20	0	20	0	0	280	8:10:20	3:30:20
11	19	Marek	16:01:24	82	77	0	95	87	20	0	0	20	20	0	45	401	8:01:24	2:05:24
12	2	Matěj	15:04:46	90	65	0	0	94	20	20	0	0	20	20	45	329	7:04:46	2:20:46
13	30	Martin	16:01:35	90	113	0	101	83	20	20	0	20	20	0	135	467	8:01:35	2:29:35

Obrázek 3.19: Výsledky z roku 2018, upraveno

3.5 Finální podoba zařízení a jeho otestování v praxi

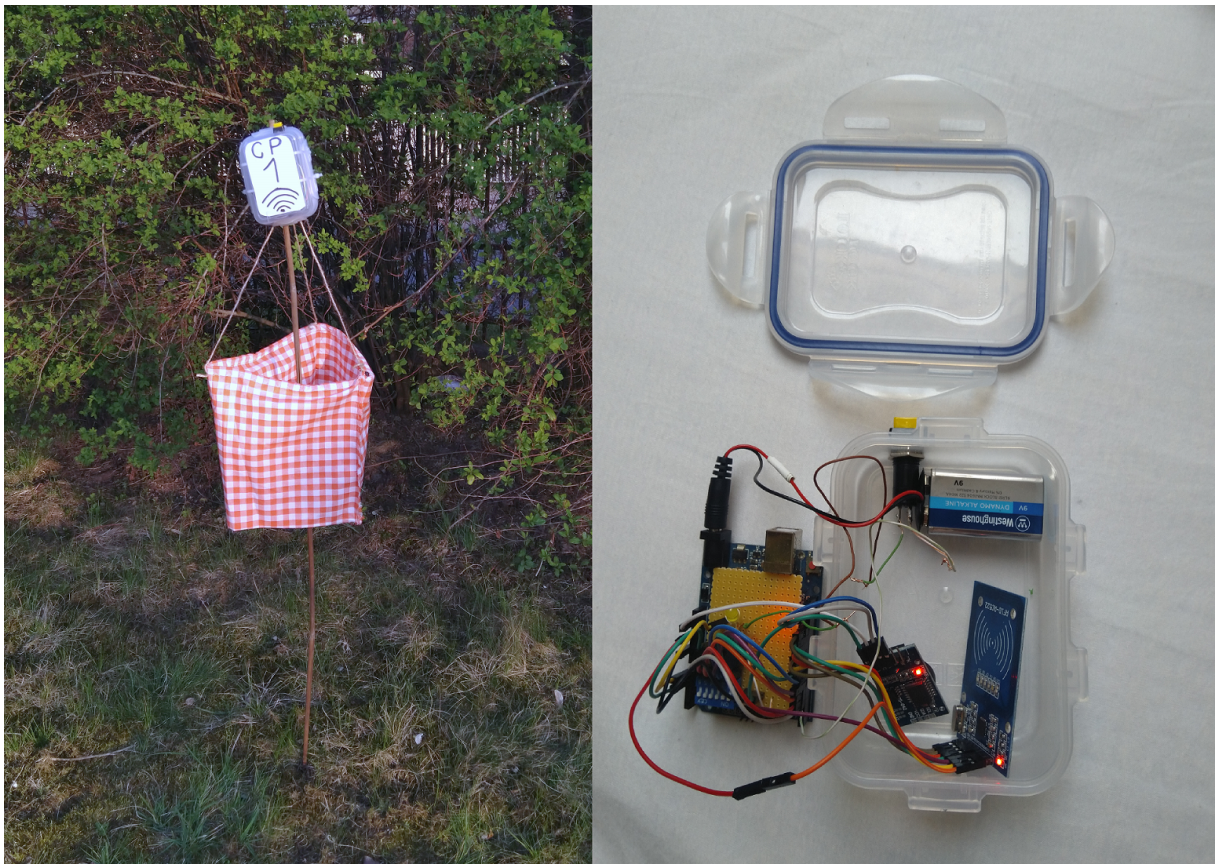
Vytvořený prototyp je pouze souhrn modulů odpovídajících požadavkům na vytvoření systému pro vyhodnocení výsledků závodu. Některé komponenty nebyly pořizovány jen s ohledem na cenu, ale také dle své dostupnosti. Je velmi pravděpodobné, že podoba výsledného zařízení bude rozdílná od prototypu. Ať už kvůli co nejnižším nákladům, anebo z příliš vysokého výkonu zařízení, který není potřeba.

3.5.1 Finální podoba

K návrhu a konstrukci zařízení patří i její podoba a umístění v terénu. Proto je potřeba vymyslet, jak bude vypadat obal, ve kterém bude zařízení ukryto, a dále pak i vhodný systém upevnění.

Pro zamezení kontaktu elektroniky s vodou byla zvolena jídelní dóza Lock & Lock 350 ml, díky své vysoké vzduchotěsnosti. Její rozměry jsou: 12,8 x 9,4 x 5,1 cm. Do této krabičky bylo potřeba vyřezat otvor na tlačítko.

Stojan se skládá z dřevěné tyčky, lampionu a stahovacích pásek. Stojan musí být dostatečně dlouhý, aby i po ukotvení v zemi bylo zařízení v dostatečné výšce. Výrazná barva lampionu zajistí, že stojan se zařízením bude rychle nalezitelný. Pro samotné ukotvení zařízení ke stojanu jsou použity stahovací pásky.



Obrázek 3.20: Vlevo - Fotografie zařízení v terénu, vpravo - Fotografie komponentů v krabičce

3.5.2 Kalkulace nákladů

Původní odhadované náklady byly stanoveny na 500 Kč za jedno zařízení. Což v případě, kdyby byly součástky kupované v českých obchodech, by pořizovací cena vyšla dvakrát více. Naštěstí je možné nakupovat i na zahraničních serverech, kde tyto součástky je možné pořídit za mnohem nižší cenu. Navíc většina zahraničních obchodníků si neúčtuje poplatek za dopravu. Většina modulů a mikroprocesor byla zakoupena v Číně, krabička a další drobné elektronické součástky potom v tuzemských obchodech.

I v případě, kdy byl použit pro návrh klon Arduina místo originálu, je z uvedené tabulky patrné, že jde o nejdražší položkou. Mezi další významné položky z hlediska ceny patří plastová krabička a baterie. Ve výsledné částce není započítaná cena za stahovací pásky, látku na lampion a další. Jejich hodnota je zanedbatelná.

Tabulka 3.5: Náklady na prototyp rok 2019

Položka	Cena v [Kč]
Klon Arduino UNO	80,-
RC522	25,-
DS3231	20,-
DIP switch	3,7
LED dioda	0,5
Piezo bzučák	2,-
Tlačítko	5,-
Krabička	45,-
Nepájivé propojovací pole	8,-
Propojovací kabely	5,-
Konektor na baterku	5,-
Baterie	45,-
Celkem	244,2

3.5.3 Možnosti vylepšení

V návrhu je použité Arduino UNO napájené z 9V baterie. Bylo zjištěno, že toto řešení není nejvhodnější. UNO má jednak vysoký výkon, který v našem zařízení nevyužijeme, a dále tato základní deska obsahuje příliš mnoho komponentů. S výkonem je spojená i vysoká spotřeba. Pro naše potřeby je mnohem vhodnější použít Arduino Mini Pro.

Na obě provedení Arduina byl nahrán stejný program. RFID čtečka vysílá jen v určitých intervalech. Rozdíl mezi spotřebou Arduina v případě, kdy anténa vysílá, nebo ne, není zanedbatelný. U Una je to při vysílání 97,94 mA, 52,3 mA, v případě když anténa nevysílá, a 39,19 mA, když je Arduino v režimu spánku. Mini má přibližně poloviční spotřebu. Při aktivní anténě spotřebuje 50,01 mA. Mezi neaktivní anténou a režimem spánku nebyl moc velký rozdíl. V obou případech bylo naměřeno 25,45 mA. V příloze je pomocí osciloskopu změřené vstupní napětí pro UNO - příloha C a Mini Pro - příloha D. Pro měření byl použit odpor o velikosti 1 Ω.

ZÁVĚR

Cílem této práce bylo vytvoření prototypu zařízení, které bude sloužit k ověření faktu, že účastníci hry „Copak je to za vojáka“ doběhli na stanoviště. Dále byl vytvořen software, který vyhodnocuje výsledky této hry.

V prvních dvou kapitolách je teoreticky popsána technologie RFID a platforma Arduino. Ve třetí kapitole je popsán návrh systému po hardwarové i softwarové stránce. Jsou zde uvedeny jednotlivé moduly, které jsou použity pro konstrukci zařízení a popsány části kódu. V této kapitole je popsán proces pro přenos dat z čipů do počítače a následný přepočet těchto dat pro zjištěni výsledků hry.

V průběhu realizace a testování zařízení nastaly problémy. Jeden z nich se objevil již při vývoji zařízení. Jedná se o občasnou nefunkčnost RFID antény. Projevovalo se to tím, že čtečka nebyla schopná navázat komunikaci s přiloženým čipem. Ověřené řešení tohoto problému spočívalo v provedení restartu zařízení. Toto řešení ale v terénu není možné provést. Znamená to, že je protřeba provést softwarový reset Arduina, který se provede po stisknutí tlačítka, které je umístěno na zařízení.

Celý systém se podařilo úspěšně realizovat ve formě funkčního a plně použitelného prototypu, čímž byly všechny body zadání splněny. Samotný vývoj zařízení byl po mě velice přínosný. I proces testování byl velmi zajímavý a vnesl do problematiky vývoje zařízení mnoho nových hodnotných informací a postupů. Tento systém bude využit na podzim tohoto roku, ale do té doby je potřeba vytvořit okolo 30 zařízení.

Systém má po menší modifikaci univerzální využití a hodí se i pro menší klasické běžecké závody s hromadným startem všech běžců nebo i postupným. Za zmínku stojí také to, že při větší modifikaci by bylo také možné na čip zapisovat všechny informace (body ze stanovišť, počet ztracených životů) a nebylo by potřeba tyto údaje přepisovat z papírů do počítače.

Literatura

[1] Zheng, F. – Kaiser, T. *Digital signal processing for RFID.* Chichester, West Sussex, United Kingdom: Wiley, 2016. ISBN 9781118824313.

[2] Roberti, M. *The History of RFID Technology.* p. 2. URL: <https://www.rfidjournal.com/articles/view?1338/2> (cit. 2019-03-30).

[3] Li, C. *Automatic vehicle identification (AVI) system based on RFID.* In 2010 International Conference on Anti-Counterfeiting, Security and Identification, pp. 281–284. IEEE, 2010. ISBN 978-1-4244-6731-0. doi:10.1109/ICASID.2010.5551336. URL: <http://ieeexplore.ieee.org/document/5551336/> (cit. 2019-03-31).

[4] *RFID card.* URL: <http://www.rfidcardchina.com/79-thickbox_default/original-nxp-mifare-classic-1k-card-printing.jpg> (cit. 2019-04-01).

[5] *RFID – Identifikace budoucnosti již dnes.* URL: <http://bartech.cz/reseni/technologie-rfid/> (cit. 2019-04-01).

[6] ČERNÝ, T. *Technologie RFID: možnosti jejich využití a nasazení v podniku.* Praha: Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Katedra informačních technologií, 2007.

[7] Dobkin, D. M. *The RF in RFID.* Amsterdam: Elsevier/Newnes, 2013, second edition. ed. ISBN 978-0-12-394583-9.

[8] Smiley, S. *Active RFID vs. Passive RFID: What's the Difference?* URL: <https://blog.atlasrfidstore.com/active-rfid-vs-passive-rfid> (cit. 2018-12-12).

[9] Vojtěch, L. *RFID - technologie pro internet věcí.* URL: <http://access.feld.cvut.cz/view.php?nazevclanku=rfid-technologie-pro-internet-veci&cisloclanku=2009020001> (cit. 2018-12-09).

[10] Swedberg, C. *USDA Releases RFID Animal-Tracking Project Report.* RFID Journal, p. 2. URL: <https://www.rfidjournal.com/articles/view?3303/2> (cit. 2018-12-16).

[11] Violino, B. *New "Pay As You Go"RFID System.* RFID Journal, p. 1. URL: <https://www.rfidjournal.com/articles/view?440> (cit. 2019-04-01).

[12] Pešek, D. *RFID - radiofrekvenční identifikace: důvod k obavám?* Praha: Sdružení českých spotřebitelů, c2010, 1. vyd ed. ISBN 978-80-903930-9-7.

[13] Šimera, D. *Autentizace v systému eLogika pomocí NFC*, 2014. URL: <http://hdl.handle.net/10084/103977> (cit. 2019-04-01).

[14] Novak, M. – Kalova, J. – Pech, J. *Use of the Arduino Platform in Teaching Programming.* 2018 IV International Conference on Information Technologies in Engineering Education (Inforino), pp. 1–4, 2018. doi:10.1109/INFORINO.2018.8581788. URL: <https://ieeexplore.ieee.org/document/8581788/> (cit. 2019-03-13).

[15] *Arduino shields.* URL: <https://ahmed.hotglue.me/?arduino.head.138646010684> (cit. 2019-04-01).

[16] *Arduino - Board Description.* URL: <https://www.tutorialspoint.com/arduino/arduino_board_description.htm> (cit. 2019-04-01).

[17] Svoboda, A. *Vše o napájení Arduina.* URL: <https://navody.arduino-shop.cz/technikuv-blog/napajeni-arduina.html> (cit. 2019-03-31).

[18] Selecký, M. *Arduino.* Brno: Computer Press, 2016, 1. vydání ed. ISBN 978-80-251-4840-2.

[19] *Externí sériové sběrnice SPI a I2C.* URL: <https://www.root.cz/clanky/externi-seriove-sbernice-spi-a-i2c/> (cit. 2019-03-07).

[20] Ramzan, A. – Rehman, S. – Perwaiz, A. *RFID technology.* In 2017 2nd International Conference on Control and Robotics Engineering (ICCRE), pp. 189–193. IEEE, 2017. ISBN 978-1-5090-3774-2. doi:10.1109/ICCRE.2017.7935068. URL: <http://ieeexplore.ieee.org/document/7935068/> (cit. 2019-03-31).

[21] Johnson, D. *Implementing serial bus interfaces with general purpose digital instrumentation.* In 2009 IEEE AUTOTESTCON, pp. 125–129. IEEE, 2009. ISBN 978-1-4244-4980-4. doi:10.1109/AUTEST.2009.5314057. URL: <http://ieeexplore.ieee.org/document/5314057/> (cit. 2019-03-31).

[22] *Čtečka RFID.* URL: <http://www.hwpro.cz/oc/index.php?route=product/product&product_id=235> (cit. 2019-03-27).

[23] *Arduino RFID čtečka s vestavěnou anténou.* URL: <https://arduino-shop.cz/arduino/833-arduino-rfid-ctecka-s-vestavenou-antenou-1500635997.html> (cit. 2019-02-27).

[24] *What is RFID? How It Works? Interface RC522 RFID Module with Arduino.* URL: <https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/> (cit. 2019-03-01).

[25] Koukolík, V. *Využití piezoelektrického jevu v praxi.* Plzeň: Západočeská univerzita v Plzni, 2013. URL: <http://hdl.handle.net/11025/10094>.

[26] VODA, Z. *TINYLAB: PIEZO BZUČÁK.* 2017. URL: <https://arduino.cz/tinylab-piezo-bzucak/> (cit. 2019-03-13).

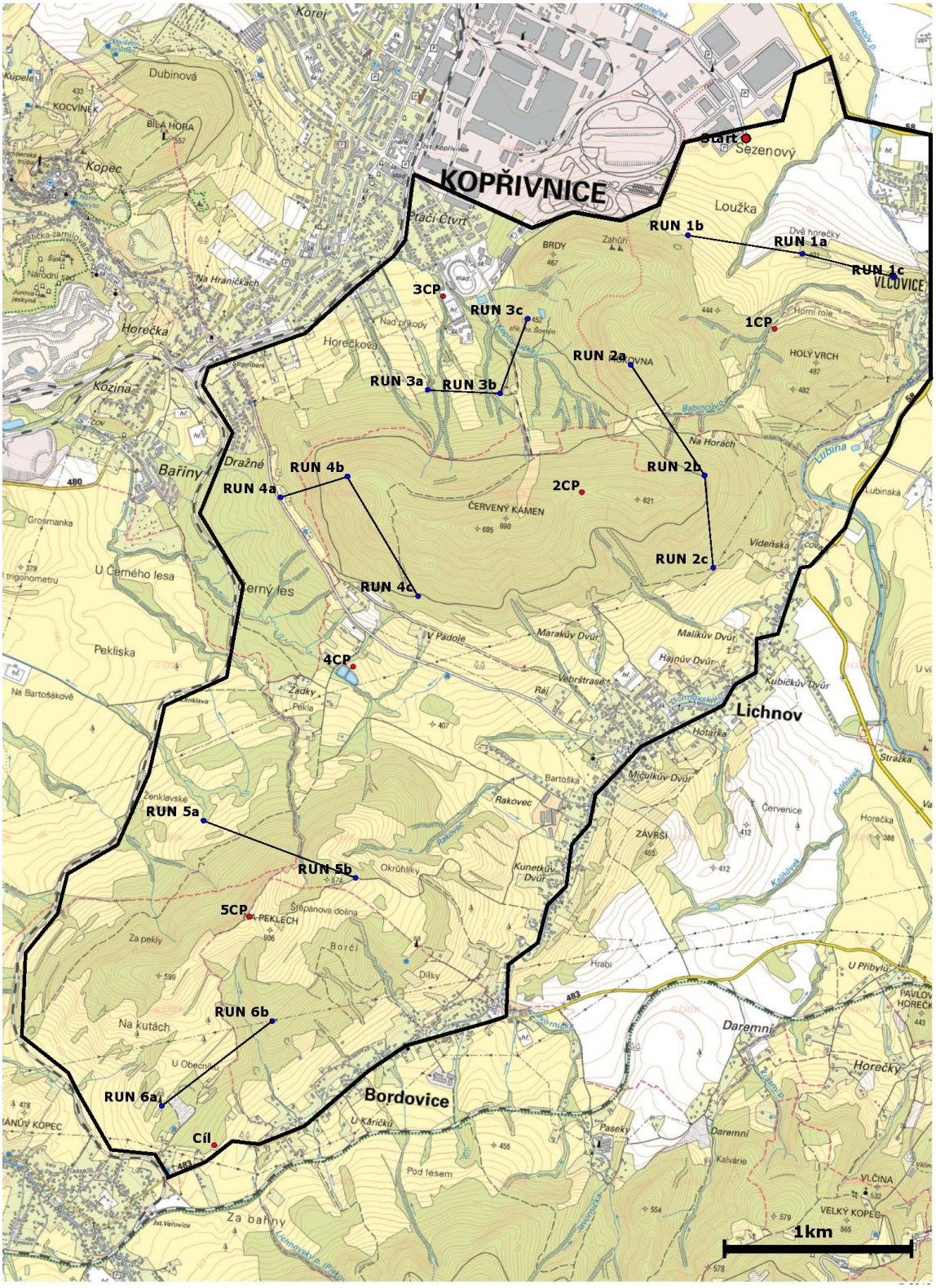
[27] Truhlář, J. *Inteligentní přístupový systém pro větší objekty*. Brno: Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav počítačových systémů, 2017.

[28] Kurk, A. *Tutorial: a guide to putting your arduino to sleep*. URL: <https://thekurks.net/blog/2018/1/24/guide-to-arduino-sleep-mode> (cit. 2019-03-26).

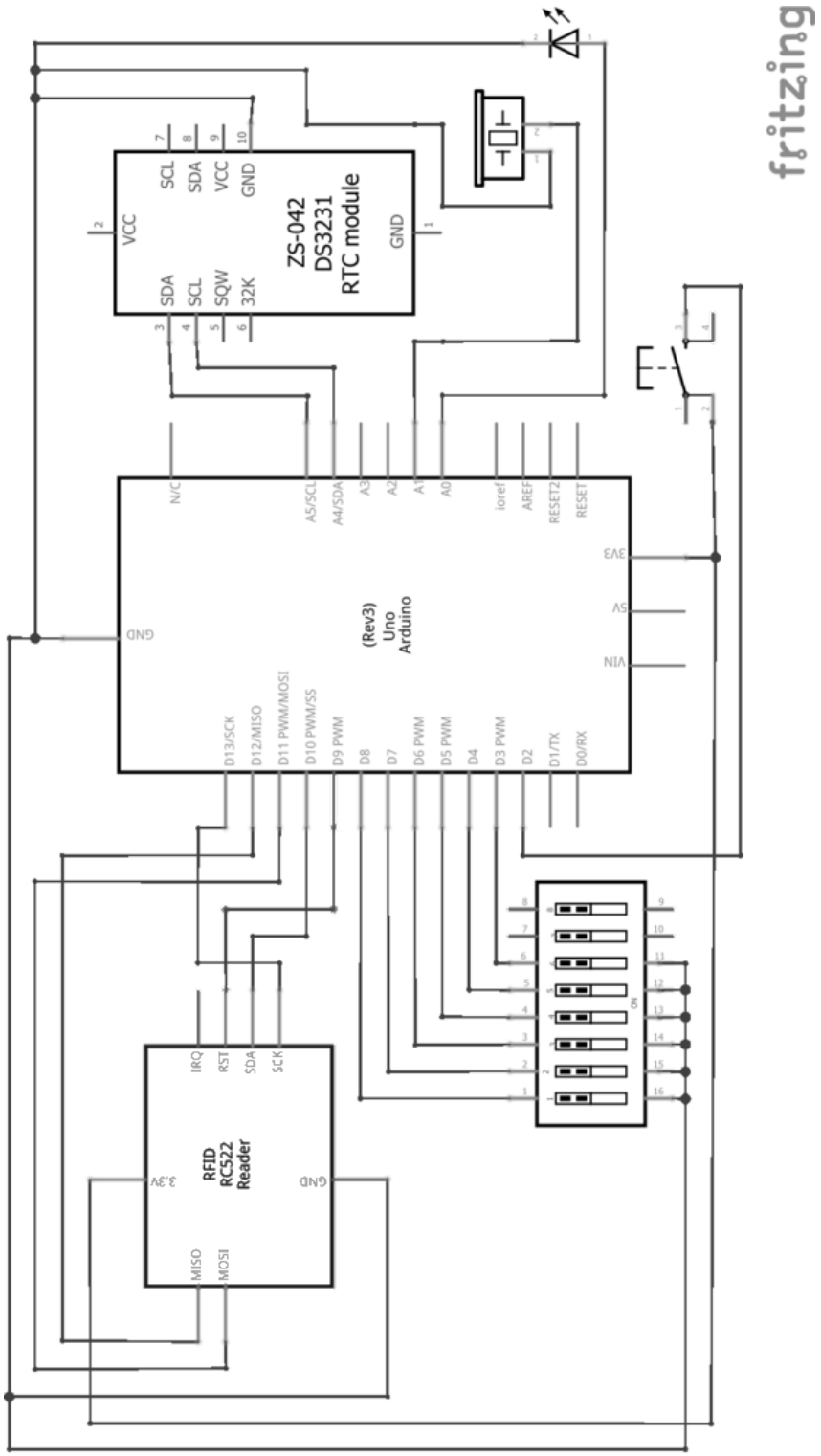
[29] *External Interrupts*. URL: <https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/> (cit. 2019-03-26).

[30] *PLX-DAQ*. URL: <https://www.parallax.com/downloads/plx-daq> (cit. 2019-03-30).

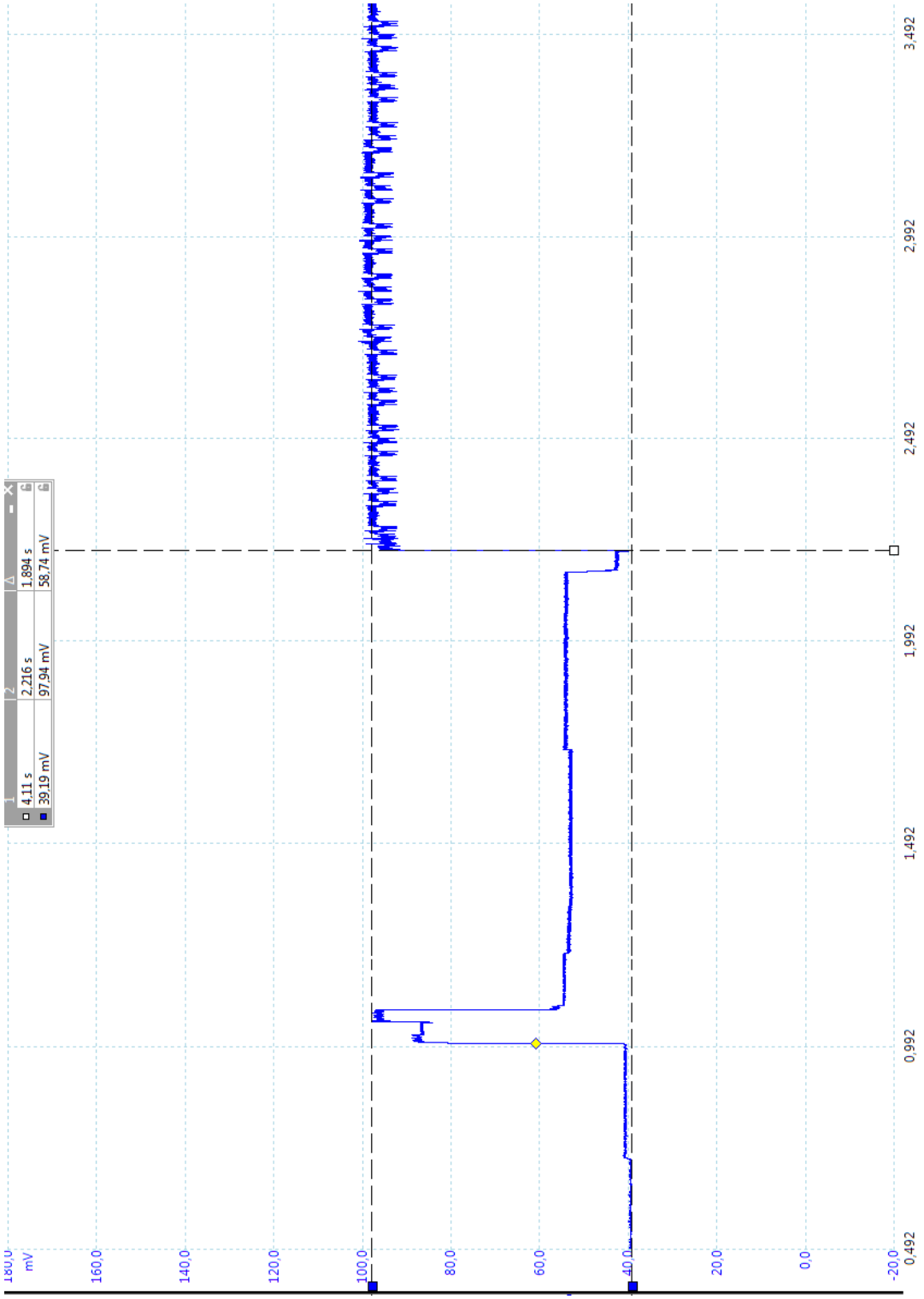
A Ukázka mapy herního území



B Schéma zapojení navrhovaného zařízení



C Spotřeba Arduino UNO



D Spotřeba Arduino Mini Pro